

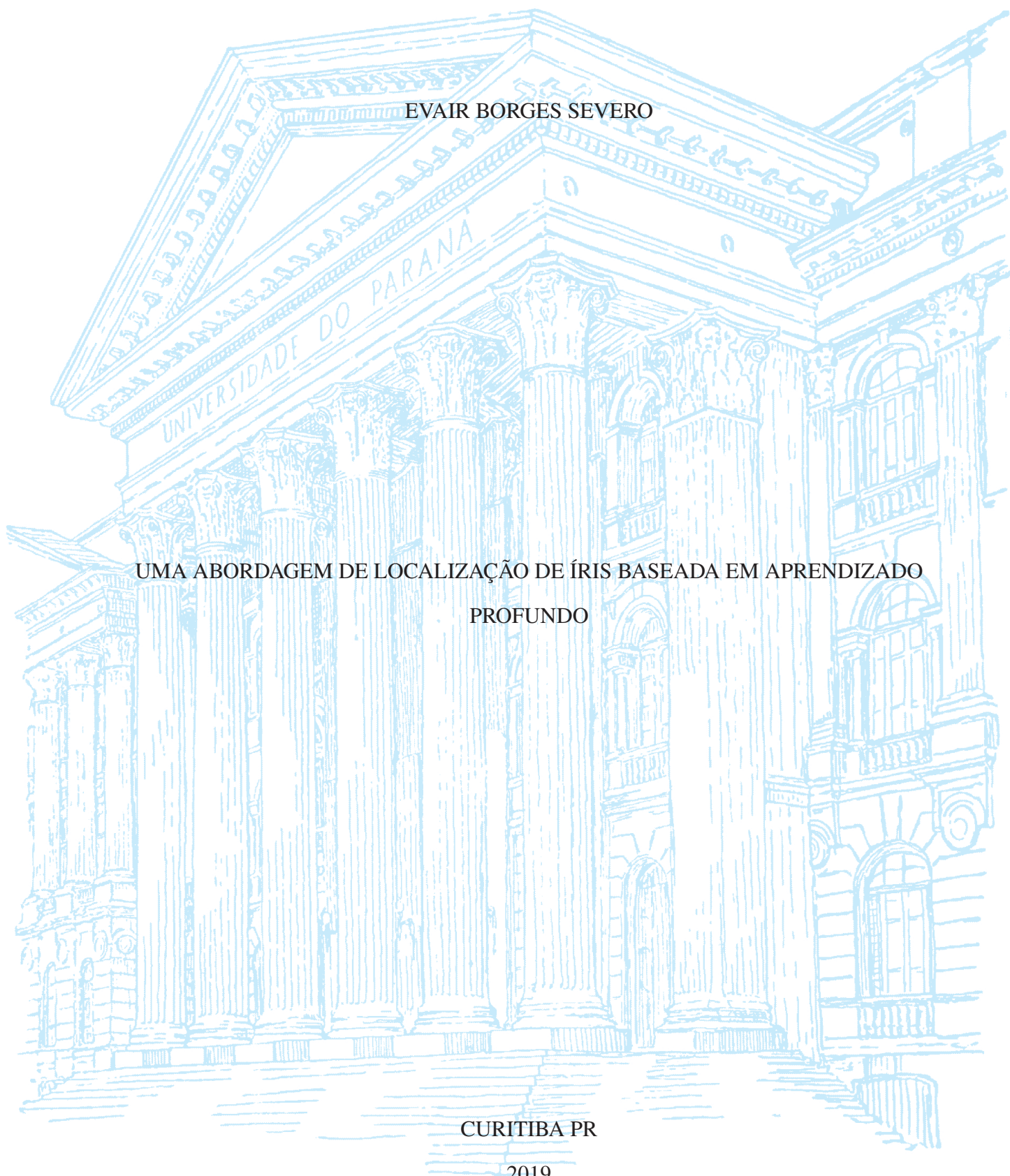
UNIVERSIDADE FEDERAL DO PARANÁ

EVAIR BORGES SEVERO

UMA ABORDAGEM DE LOCALIZAÇÃO DE ÍRIS BASEADA EM APRENDIZADO
PROFUNDO

CURITIBA PR

2019



EVAIR BORGES SEVERO

UMA ABORDAGEM DE LOCALIZAÇÃO DE ÍRIS BASEADA EM APRENDIZADO
PROFUNDO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: David Menotti.

CURITIBA PR

2019

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

- S498a Severo, Evair Borges
Uma abordagem de localização de íris baseada em aprendizado profundo [recurso eletrônico] / Evair Borges Severo – Curitiba, 2019.
- Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática.
Orientador: David Menotti
1. Iris – Identificação biométrica. 2. Localização de Íris. I. Universidade Federal do Paraná. II. Menotti, David. III. Título.

CDD: 570.15195

Bibliotecária: Roseny Rivelini Morciani CRB-9/1585



MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA -
40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **EVAIR BORGES SEVERO** intitulada: **UMA ABORDAGEM DE LOCALIZAÇÃO DE ÍRIS BASEADA EM APRENDIZADO PROFUNDO**, sob orientação do Prof. Dr. DAVID MENOTTI GOMES, que após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 23 de Setembro de 2019.



DAVID MENOTTI GOMES

Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)



LUCAS FERRARI DE OLIVEIRA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)



YANDRE MALDONADO E GOMES DA COSTA

Avaliador Externo (UNIVERSIDADE ESTADUAL DE MARINGÁ)



LUIZ EDUARDO SOARES DE OLIVEIRA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)



*Aos meus pais, Francisco e Iolanda,
por terem abdicado de suas vidas em
prol das realizações e da felicidade
de seus filhos.*

*À minha amada esposa
Rafaela, por ser uma luz e uma
fonte de motivação em minha vida,
por todo amor e compreensão que
demonstra diariamente.*

AGRADECIMENTOS

Em primeiro lugar e acima de tudo, agradeço à Deus, meu Pai Celestial, por ter me guiado no caminho que me trouxe até aqui. Obrigado pela oportunidade de viver nesta terra e, assim, obter conhecimento e me desenvolver. Obrigado por Seu infinito amor e por ser totalmente acessível a todos que o buscam.

Sou grato aos meus pais, Francisco e Iolanda, por todo o amor e dedicação que demonstraram durante toda a minha vida. Por terem me apoiado e proporcionado condições de chegar até aqui. Sinto-me orgulhoso e privilegiado por ter pais tão especiais.

Agradeço à minha esposa, Rafaela, por ser uma fonte de luz, motivação e amparo na minha vida. Obrigado por todo amor, compreensão e paciência em tantos momentos difíceis desta caminhada.

Ao meu orientador, Professor David, que por meio de seu conhecimento, incentivo e experiência, foi decisivo para que esse trabalho se concretizasse. Serei eternamente grato pelo seu incessante encorajamento e por ter acreditado no meu potencial.

À todos colegas/amigos que tive a oportunidade de conhecer durante a graduação e mestrado, pela convivência, apoio e aprendizado que adquiri com todos.

RESUMO

A íris é considerada uma das características biométricas que possui maior grau de unicidade. A localização automática da íris consiste em uma importante tarefa em sistemas biométricos, podendo afetar diretamente os resultados obtidos por aplicações específicas, tais como reconhecimento de íris, detecção de lentes de contato, *spoofing*, dentre outras. Esse trabalho define o problema de localização da íris como a delimitação, por meio da menor janela quadrada, que engloba a região da íris. Nesse contexto, o objetivo geral deste trabalho consiste em propor um método baseado em *deep learning* para a localização de íris em imagens da região periocular. Por meio desse objetivo, pretende-se obter um detector de íris aplicável em diferentes bases de imagens, em ambientes cooperativos e não cooperativos, superando os desafios e limitações observados nos métodos propostos na literatura. Utilizamos como arcabouço de aprendizagem profunda a Darknet, que implementa uma rede neural *open source* utilizada pela YOLO (*You Only Look Once*), sendo estado da arte em detecção de objetos em tempo real. Para o aprendizado e avaliação de tais detectores, é necessário ter, além das bases de dados, as anotações das *bounding boxes*, isto é, o *groundtruth*, nas imagens utilizadas. Portanto, a tarefa de criar as anotações manuais das bases de dados consiste em um dos objetivos específicos deste trabalho. Foram criadas as anotações de mais de 10 000 imagens, pertencentes à sete bases de dados. Tais anotações estão disponíveis publicamente e podem ser utilizadas para fins científicos. Adicionalmente, com o propósito de criar um *baseline* para detectores de íris, foi proposto um detector baseado em janela deslizante, usando características extraídas por *Histogram of Oriented Gradients* (HOG) e *Support Vector Machine* (SVM) sendo utilizado como classificador, ou seja, uma adaptação do método de detecção de pedestres proposto por Dalal & Triggs. Os experimentos mostram que o uso do detector de objetos YOLO, baseado em *Deep Learning*, aplicado para a localização de íris apresenta resultados promissores em todas as bases de dados utilizadas. Nos experimentos em que a YOLO foi testada em bases “desconhecidas”, em mais de 90% das imagens testadas foi reportado um valor de *IoU* superior à 80%. Tal observação também é válida para as métricas *recall* e *precision*.

Palavras-chave: Localização de Íris, HoG & Linear SVM, Darknet YOLO, Aprendizado Profundo

ABSTRACT

The iris is considered as the biometric trait with the highest unique probability. The iris location is an important task for biometrics systems, affecting directly the results obtained in specific applications such as iris recognition, spoofing and contact lenses detection, among others. This work defines the iris location problem as the delimitation of the smallest squared window that encompasses the iris region. In this context, the general objective of this work is to propose a deep learning method for iris location in periocular region images. Through this objective, it is intended to obtain an iris detector applicable in different databases, in cooperative and non-cooperative environments, overcoming the limitations observed in the methods proposed in the literature. As a deep learning approach, we use the Darknet, which implements an open source neural network used by YOLO, being state of the art in real time object detection. Additionally, for the purpose of creating a baseline for iris detectors, we also propose a sliding window detector based on features from Histogram of Oriented Gradients (HOG) and a linear Support Vector Machines (SVM) classifier, i.e., an adaptation from the human detection method proposed by Dalal & Triggs. For the learning and evaluation of such detectors, it is necessary to have, besides the databases, the annotations of bounding boxes (groundtruth). Therefore, the task of creating manual database annotations is one of the specific objectives of this work. Annotations of over 10 000 images from seven databases were created. These annotations are publicly available and may be used for scientific purposes. Experiments show that the use of the Deep Learning-based YOLO object detector for iris location has shown promising results in all databases used. In experiments where YOLO was applied to “unknown” databases, *IoU* greater than 80% was reported in more than 90% of the images tested. This observation is also valid for the recall and precision metrics.

Keywords: Iris location, HOG & Linear SVM, Darknet YOLO, Deep Learning

LISTA DE FIGURAS

1.1	Principais estruturas observadas em um olho humano..	12
1.2	Exemplos de imagens da região periocular..	13
1.3	Localização da íris e sua normalização.	13
2.1	Exemplo de imagem descrita por HOG. Imagem obtida da base IIIT-Delhi Contact Lens Iris (Kohli et al., 2013)	17
2.2	Hiperplano gerado pelo SVM.	18
2.3	Ilustração de um modelo de aprendizado profundo.	19
2.4	Exemplo das camadas de uma Rede Neural Convolucional.	20
2.5	Deteção de objetos usando YOLO. Fonte: (Redmon, 2016b).	21
2.6	Arquitetura da YOLO. Possui 24 camadas convolucionais seguidas de 2 camadas totalmente conectadas. Camadas convolucionais alternadas de 1×1 reduzem o espaço de características das camadas anteriores. As camadas convolucionais são pré-treinadas na metade da resolução (224×224) e, em seguida, no dobro da resolução para detecção. Imagem reproduzida de (Redmon et al., 2016a).	22
2.7	Exemplo de detecção de objetos com (a) Fast-YOLO e (b) YOLO. Imagem replicada de https://pjreddie.com/darknet/yolo	23
2.8	Ilustração de uma das limitação da YOLO, em que a detecção pode perder objetos muito próximos. Em (a), há 9 indivíduos no canto inferior esquerdo, porém apenas 5 foram detectados, como pode ser visto em (b). Imagem replicada de (Redmon et al., 2016a).	23
2.9	Exemplo de <i>anchor boxes</i>	24
3.1	Proposta de Daugman para mitigar o efeito de oclusões superiores e inferiores. Fonte: (Oad e Ahmad, 2012)	26
3.2	Pontos e vértices definidos no método de Rodríguez e Rubio, 2005. Fonte: (Rodríguez e Rubio, 2005)	27
3.3	Pontos a serem detectados no método de Su et al. (2017). Fonte: (Su et al. 2017).	30
4.1	Exemplos de imagens encontradas na base de dados <i>IIIT-Delhi Contact Lens Iris</i>	34
4.2	Exemplos de imagens encontradas na base de dados <i>CASIA-IrisV3 - Interval</i>	35
4.3	Exemplos de imagens encontradas na base de dados <i>Notre Dame Cosmetic Contact Lenses</i>	35
4.4	Exemplos de imagens encontradas na base de dados <i>Notre Dame Contact Lens Detection 2015</i>	36
4.5	Exemplos de imagens encontradas na base de dados BERC.. . . .	36
4.6	Exemplos de imagens encontradas na base de dados <i>CASIA-IrisV2</i>	37
4.7	Exemplos de imagens encontradas na base de dados <i>MobBIOfake</i>	38

4.8	Exemplos de imagens encontradas na base de dados MICHE-I, obtidas com <i>iPhone 5</i>	38
4.9	Exemplos de imagens encontradas na base de dados MICHE-I, tomadas com <i>Samsung Galaxy S4</i>	39
4.10	Exemplos de imagens encontradas na base de dados MICHE-I, obtidas com <i>Samsung Galaxy Tablet II</i>	39
4.11	Ilustração de pixels Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos, definindo a região de <i>ground truth</i> e região detectada.	39
4.12	Desbalanceamento de classes observado em imagens da região periocular: Classe positiva (íris) delimitada em verde e classe negativa delimitada em vermelho, ou seja, a região da íris é, geralmente, uma pequena porção da imagem. Imagem obtida da base MICHE-I (De Marsico et al., 2015).	41
4.13	Exemplos de amostras positivas utilizadas para extração de características e treinamento do SVM.	42
4.14	Exemplos de amostras negativas utilizadas para extração de características e treinamento do SVM.	42
5.1	Exemplos de detecções (em bases desconhecidas) em imagens com significativa presença de ruídos	47
5.2	Curvas das métricas Recall, Precision e IoU para os resultados obtidos pela YOLO em experimentos com bases desconhecidas VIS (Tabela 5.5).	48
5.3	Curvas das métricas Recall, Precision e IoU para os resultados obtidos pela YOLO em experimentos com bases desconhecidas NIR (Tabela 5.5).	49
5.4	Exemplos de detecções obtidas pela YOLO e seus respectivos valores de IoU. A <i>bounding box</i> verde refere-se ao <i>ground truth</i> e a <i>bounding box</i> vermelha refere-se à região detectada.	49

LISTA DE TABELAS

3.1	Análise sumarizada dos trabalhos relacionados.	31
4.1	Principais informações das bases de imagens NIR	34
4.2	Principais informações das bases de imagens VIS	37
4.3	Arquitetura da rede <i>Fast-YOLOv3</i>	43
5.1	Resultados <i>Intra-sensor</i> - Bases de dados NIR (%)	44
5.2	Resultados <i>Intra-sensor</i> - Bases de dados VIS (%).	45
5.3	Resultados <i>Inter-sensor</i> (%)	45
5.4	Resultados da combinação de sensores (%)	46
5.5	Avaliação de robustez da YOLO - Teste em bases desconhecidas (%)	46

LISTA DE ACRÔNIMOS

Acc	<i>Accuracy</i>
BERC	<i>Biometrics Engineering Research Center</i>
CNN	<i>Convolutional Neural Network</i>
FN	Falso Negativo
FP	Falso Positivo
HOG	<i>Histogram of Oriented Gradients</i>
IIIT-D CLI	<i>IIIT-Delhi Contact Lens Iris</i>
IoU	<i>Intersection over Union</i>
mAP	<i>Mean Average Precision</i>
MICHE	<i>Mobile Iris Challenge Evaluation</i>
MMQ	Método dos Mínimos Quadrados
NDCCL	<i>Notre Dame Cosmetic Contact Lenses</i>
NDCLD15	<i>Notre Dame Contact Lens Detection 2015</i>
NIR	<i>Near-Infrared</i>
Prec	<i>Precision</i>
OWA	<i>Ordered Weighted Averaging</i>
Rec	<i>Recall</i>
ReLU	<i>Rectified Linear Unit</i>
SGS4	<i>Samsung Galaxy S4</i>
SGTII	<i>Samsung Galaxy Tab II</i>
SVM	Support Vector Machine
VIS	<i>Visible</i>
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	DESCRIÇÃO DO PROBLEMA	13
1.2	OBJETIVOS	14
1.3	CONTRIBUIÇÕES	14
1.4	ORGANIZAÇÃO DO DOCUMENTO	15
2	FUNDAMENTAÇÃO TEÓRICA.	16
2.1	RECONHECIMENTO DE PADRÕES	16
2.1.1	Histogram of Oriented Gradients (HOG).	16
2.1.2	Support Vector Machine (SVM)	17
2.2	APRENDIZADO PROFUNDO	18
2.2.1	Redes Neurais Convolucionais	19
2.3	YOLO - DETECÇÃO DE OBJETOS COM ABORDAGEM DE APRENDIZADO PROFUNDO	21
2.3.1	YOLOv2	24
3	TRABALHOS RELACIONADOS	25
3.1	TRABALHOS RELACIONADOS A LOCALIZAÇÃO DE ÍRIS	25
3.2	ANÁLISE DOS TRABALHOS RELACIONADOS À LOCALIZAÇÃO DE ÍRIS	30
4	MATERIAIS E MÉTODOS	33
4.1	BASES DE DADOS	33
4.1.1	Bases de Imagens NIR	33
4.1.2	Bases de Imagens VIS	37
4.2	MEDIDAS DE AVALIAÇÃO.	38
4.3	ABORDAGENS PROPOSTAS	41
4.3.1	Janelas deslizantes com HOG e SVM	41
4.3.2	<i>Fast-YOLOv3</i>	43
5	EXPERIMENTOS.	44
5.1	CONFIGURAÇÃO EXPERIMENTAL.	44
5.2	RESULTADOS E DISCUSSÃO	44
6	CONCLUSÕES	50
6.1	PUBLICAÇÕES	50
	REFERÊNCIAS	52

1 INTRODUÇÃO

A biometria é a ciência da identificação de indivíduos com base em características físicas ou comportamentais, tais como impressões digitais, face, voz, íris, anatomia das mãos, dentre outras (Chauhan et al., 2010; Le e Jain, 2009; Raghavendra et al., 2017). Os sistemas biométricos têm provido um significativo avanço na identificação e autenticação de pessoas, exercendo um papel importante na segurança pessoal, nacional e global (Menotti et al., 2015). Consiste em uma área de pesquisa em grande crescimento nos últimos anos, podendo abranger a computação, engenharia elétrica, forense e estatística, com diversas oportunidades de pesquisa, que são alimentadas pela necessidade de maior segurança e uso eficiente dos sistemas biométricos (Crihalmeanu et al., 2007).

Dentre as características físicas mencionadas, a íris aparece como uma das mais representativas e seguras modalidades biométricas (Turk e Pentland, 1991; Daugman, 1993; Wildes, 1997; Jain et al., 2004; Proença e Alexandre, 2006). Tal observação pode ser feita pois a íris consiste em uma região do olho humano que permite distinguir indivíduos com alto grau de unicidade (Zhu et al., 2000). Atualmente, essa modalidade biometrica é utilizada em diversas áreas, tais como controle de fronteiras internacionais, transações financeiras e segurança computacional. Além disso, o processo de identificação não é invasivo, ou seja, não há a necessidade de contato físico com a íris para que a imagem seja obtida e analisada (Jain et al., 2006).

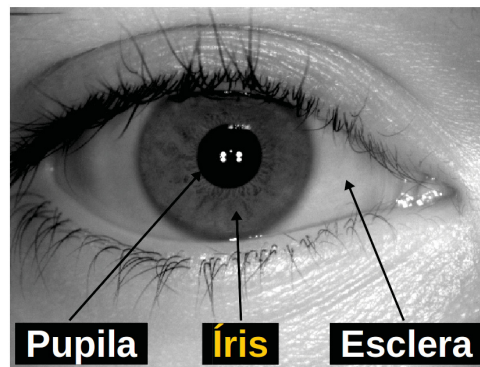


Figura 1.1: Principais estruturas observadas em um olho humano.

Na Figura 1.1, a íris pode ser observada, assim como as demais estruturas presentes em um olho humano. Daugman (1993) relata que diversos oftalmologistas observaram um alto nível de detalhes de informações e textura presentes na íris. Tais informações não sofrem alterações no decorrer do tempo de vida do ser humano (exceto em casos de doenças ou acidentes), o que caracteriza a íris como uma estrutura com um alto grau de unicidade.

Em sistemas de reconhecimento, autenticação e identificação de pessoas por meio da íris, a localização da região da íris geralmente é um dos passos iniciais e pode, consequentemente, influenciar diretamente o desempenho de tais sistemas (Wildes, 1997; Daugman, 2004; Su et al., 2017).

1.1 DESCRIÇÃO DO PROBLEMA

As imagens utilizadas como entrada em sistemas biométricos relacionados à íris normalmente correspondem a uma região em que aparece o olho e seu entorno. Essa região é conhecida como região periocular. Sendo assim, a região periocular apresenta diversas estruturas, tais como a íris, pupila, esclera, reflexos, pálpebras, cílios, cabelo e sobrancelhas (Jillela et al., 2013). Além disso, em imagens da região periocular podem aparecer acessórios como óculos, lentes de contato, maquiagem, dentre outros. A Figura 1.2 ilustra exemplos de imagens da região periocular onde pode-se observar a presença de tais estruturas e acessórios, assim como a variação significativa do tamanho efetivo da íris.

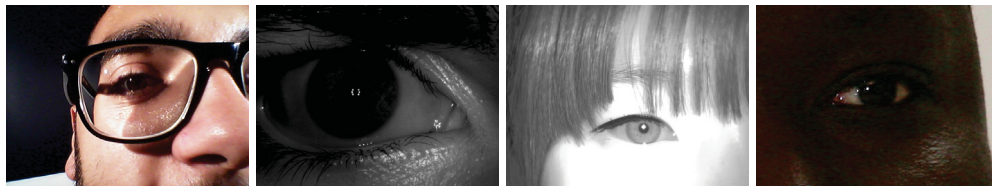
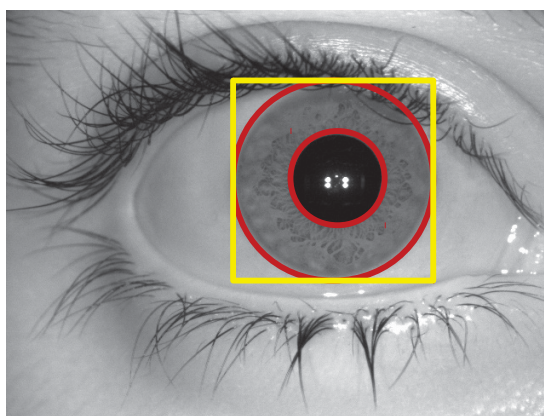


Figura 1.2: Exemplos de imagens da região periocular.

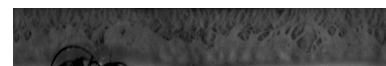
Na literatura (discutida em detalhes no Capítulo 3), são encontrados diversos trabalhos que propõem métodos de localização da íris por meio da determinação de um círculo que bem a delimite (como pode ser visualizado, em *vermelho*, na Figura 1.3(a)). Na maioria destes trabalhos, a pupila também é delineada, pois em diversas aplicações é necessário realizar a normalização da íris. A normalização consiste em transformar a região de arco da íris (entre esclera e pupila) do espaço cartesiano em um sistema de coordenadas polares, de modo que a íris seja representada por um retângulo. A Figura 1.3(b) e a Figura 1.3(c) ilustram, respectivamente, a região de arco da íris extraída e normalizada. Geralmente, as representações e características são extraídas a partir da imagem normalizada/transformada.



(a) Localização da íris usando uma *bounding box* (amarelo) e círculos (vermelho).



(b) Extração da íris por meio da localização em círculo.



(c) Íris normalizada.

Figura 1.3: Localização da íris e sua normalização.

A localização da íris pode ser considerada uma tarefa desafiadora pois, além da presença das estruturas mencionadas anteriormente, a aquisição das imagens ocorre com diferentes equipamentos e condições de iluminação, assim como variações de translação, rotação e escala (Zhu et al., 2000).

Muitos dos métodos encontrados na literatura se propõem a resolver este problema usando as informações geométricas da íris. Essas técnicas utilizam detecção de bordas, binarização,

morfologia matemática, segmentos de contornos, detecção de círculos, dentre outros (Daugman, 2004; Alvarez e Garcia, 2010; Cui et al., 2012; Su et al., 2017). Não obstante, o desempenho e precisão dos métodos encontrados na literatura são geralmente analisados em uma única base de dados e quantificados por meio de uma única métrica, o que não permite uma avaliação conclusiva sobre o correto funcionamento do método, principalmente em imagens de outras bases ou obtidas com sensores diferentes.

O crescente sucesso do uso de técnicas de aprendizado profundo (*deep learning*) por meio de redes de convolução em problemas de visão computacional (Hinton e Salakhutdinov, 2016; Pinto et al., 2011; Krizhevsky et al., 2012a; Mao et al., 2017; Fan et al., 2017; Laroca et al., 2019a)) permitiram o aprendizado de melhores representações e consequentemente a obtenção de novos resultados no estado da arte. Com isso, surge o interesse de se investigar e estudar o uso de tais técnicas para o aprendizado de representações em profundidade para problemas de biometria relacionados à íris.

Nesse contexto, como hipótese de pesquisa deste trabalho propomos que um método baseado em *deep learning* é capaz de localizar a íris com alto poder de generalização, podendo ser aplicado em diferentes bases de imagens. Além disso, o aprendizado profundo pode sobrepujar as principais limitações dos métodos encontrados na literatura (causadas, principalmente, por ruídos, diferença de ângulos, baixa qualidade da imagem e pela presença das demais estruturas que normalmente aparecem em imagens da região periocular, tais como óculos, cabelo, nariz, reflexos, etc). Este trabalho define o problema de localização da íris como a delimitação, por meio da menor janela quadrada, que engloba a região da íris, como ilustrado em *amarelo* na Figura 1.3(a).

1.2 OBJETIVOS

Conforme a discussão feita anteriormente, o objetivo geral deste trabalho consiste em propor um método baseado em aprendizagem profunda para a localização de íris em imagens da região periocular. Por meio desse objetivo, pretende-se obter um detector de íris aplicável em diferentes bases de imagens, em ambientes cooperativos e não cooperativos, superando os desafios e limitações encontrados pelos métodos propostos na literatura.

Neste trabalho, utilizamos como arcabouço de aprendizagem profunda a Darknet (Redmon, 2016a), que implementa uma rede neuronal *open source* utilizada pela YOLO (*You Only Look Once*), sendo estado da arte em detecção de objetos em tempo real (Redmon et al., 2016a,b; Redmon e Farhadi, 2018).

Para o aprendizado e avaliação de tais detectores, é necessário ter, além das bases de dados, as anotações das *bounding boxes*, isto é, o *groundtruth*, nas imagens utilizadas. Portanto, a tarefa de criar as anotações manuais das bases de dados consiste em um dos objetivos específicos deste trabalho.

Adicionalmente, com o propósito de criar um *baseline* para detectores de íris, foi proposto um detector baseado em janela deslizante, usando características extraídas por *Histogram of Oriented Gradients* (HOG) e *Support Vector Machine* (SVM) sendo utilizado como classificador, ou seja, uma adaptação do método de detecção de pedestres proposto por Dalal & Triggs (Dalal e Triggs, 2005).

1.3 CONTRIBUIÇÕES

De acordo com a definição da abordagem proposta, definem-se as contribuições esperadas por meio do desenvolvimento e conclusão deste trabalho. Dentre tais contribuições, destacam-se:

- Proposta de um localizador de íris baseado em aprendizagem profunda;
- Uma avaliação e discussão da proposta aplicada em diversas bases de imagens, por meio de quatro métricas diferentes;
- Disponibilizar publicamente ¹ — para a comunidade científica — as anotações criadas. Foram criadas as anotações de mais de 10 000 imagens, pertencentes à 7 bases de dados.
- Uma avaliação e comparação de uma abordagem baseada em HOG e SVM aplicada para localização de íris em imagens da região periocular;

Em meio a tais contribuições, dois artigos foram publicados. Primeiramente, foi realizada uma avaliação de descritores de textura e forma, aplicados para localização de íris por meio de uma abordagem de janela deslizante. As contribuições e resultados desta parte foram publicados e apresentados no XIII Workshop de Visão Computacional (WVC) (Severo e Menotti, 2017).

Além disso, uma avaliação do detector (baseado em *Deep Learning*) YOLO, aplicado na localização de íris, em forma de artigo científico foi publicada e apresentada na *International Joint Conference on Neural Networks* (IJCNN) 2018 (Severo et al., 2018).

1.4 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho está organizado da seguinte maneira: No Capítulo 2, são abordados os conceitos e métodos utilizados neste trabalho, tais como *Histogram of Oriented Gradients* (HOG), Support Vector Machine (SVM), *Convolutional Neural Networks* (CNN), dentre outros. No Capítulo 3, são abordados alguns trabalhos, encontrados na literatura, relacionados à localização de íris. No Capítulo 4, são apresentados os materiais utilizados na realização desse trabalho, bem como os métodos propostos para localização da íris em imagens da região periocular. No Capítulo 5, são apresentados e discutidos os resultados reportados nos experimentos. Por fim, no Capítulo 6, são apresentadas as conclusões e trabalhos futuros.

¹<https://web.inf.ufpr.br/vri/databases/iris-location-annotations/>.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a base teórica para dos conceitos e métodos utilizados neste trabalho. Na Seção 2.1, são apresentados os principais conceitos de Reconhecimento de Padrões, bem como uma breve abordagem acerca do descritor de características e classificador avaliados nesse trabalho. Os principais fundamentos de Aprendizado Profundo e Redes Neurais Convolucionais (CNN) são abordados na Seção 2.2. Por fim, abordamos a base teórica do detector de objetos YOLO na Seção 2.3.

2.1 RECONHECIMENTO DE PADRÕES

O Reconhecimento de Padrões tem por objetivo a classificação de objetos em um determinado número de categorias ou classes a partir da observação de suas características (Duda et al., 2003). Sendo assim, o reconhecimento de Padrões visa construir uma representação mais simples de um conjunto de dados através de suas características mais relevantes, possibilitando sua partição em classes (Theodoridis e Koutroumbas, 2003). Exemplos clássicos na aplicação do reconhecimento de padrões são: reconhecimento de escrita, reconhecimento de impressões digitais, reconhecimento de fala e reconhecimento de faces, mineração de dados, análise de imagens de documentos para reconhecimento de caracteres (*Optical Character Recognition - OCR*), dentre outros (Jain et al., 2000).

Em sistemas de reconhecimento de padrões, o estágio de extração de características (ou representação) consiste em uma importante tarefa, visto que o grau de precisão alcançado pelo método de classificação está diretamente relacionado com o nível de discriminação observado no conjunto de características extraídas. Sendo assim, o objetivo é caracterizar um objeto através de medidas, as quais são bastante similares para objetos da mesma classe, e bastante diferentes para objetos de outras classes. Entre as características visuais, pode-se citar as formas, cores, contorno, textura, dentre outras. Na sequência, serão apresentados os métodos de extração de características utilizados nesse trabalho.

2.1.1 Histogram of Oriented Gradients (HOG)

O HOG é um descritor de características utilizado em visão computacional para a detecção de objetos. Esse descritor foi primeiramente proposto por Dalal e Triggs (2005), sendo utilizado em um método de detecção de pedestres. O método quantifica as ocorrências de orientação do gradiente em regiões de uma imagem, extraíndo informações da forma dos objetos (Wang et al., 2009; Bhilare et al., 2017).

A ideia essencial apresentada pelo HOG é que, em uma imagem, a aparência e a forma de um objeto local podem ser descritas através da distribuição de gradientes de intensidade ou direções de borda. Para isso, a imagem é dividida em pequenas regiões conectadas de tamanho $C \times C$ pixels, chamadas células. Para os pixels dentro de cada célula, um histograma de direções de gradiente é calculado (Dalal e Triggs, 2005).

Para o cálculo do gradiente, o método mais comum é aplicar a máscara derivada em uma ou ambas as direções horizontal e vertical. Especificamente, esse método requer a filtragem de cor ou intensidade por meio dos seguintes *kernels*: $[-1, 0, 1]$ $[-1, 0, 1]^T$. Dalal e Triggs testaram, adicionalmente, outras máscaras, tais como o filtro de Sobel ou máscaras diagonais. Além

disso, eles também fizeram experimentos com suavização gaussiana antes de aplicar a máscara derivada.

Após ser realizado o cálculo dos histogramas de direções de cada célula, a representação é criada por meio da concatenação desses histogramas. A fim de melhorar a precisão dessas características extraídas, os histogramas locais são normalizados. Essa normalização por contraste é feita por meio do cálculo de uma medida da intensidade em uma região maior da imagem, chamada de bloco. Esse valor de intensidade é então utilizado para normalizar todas as células dentro do bloco, resultando em uma melhor invariância a mudanças na iluminação e sombreamento. A Figura 2.1 ilustra um exemplo de imagem descrita por HOG.

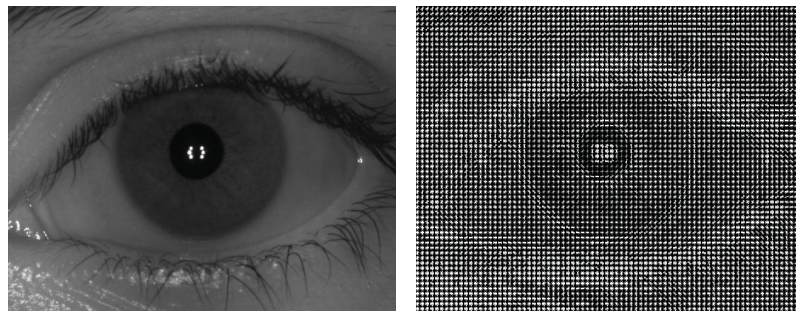


Figura 2.1: Exemplo de imagem descrita por HOG. Imagem obtida da base IIIT-Delhi Contact Lens Iris (Kohli et al., 2013)

Em diversos descritores de características, uma das primeiras etapas é garantir valores de cor e gama normalizados. Entretanto, segundo Dalal & Triggs, este passo pode ser omitido na utilização do descritor HOG, já que a normalização da representação resultante atinge, essencialmente, um resultado semelhante.

2.1.2 Support Vector Machine (SVM)

O SVM foi primeiramente apresentado por Vladimir Vapnik (Vapnik, 1995) e é um dos métodos de classificação mais utilizados em anos recentes (Franchi et al., 2016; Ruiz et al., 2014). Consiste em um método de aprendizagem supervisionada que permite analisar os dados, reconhecer padrões e classificá-los.

Basicamente, o objetivo do SVM é encontrar uma fronteira de decisão (também chamada de hiperplano) entre os exemplos das classes existentes no conjunto de treinamento. Essa fronteira de decisão é definida por meio dos vetores de suporte, como ilustrado na Figura 2.2. Portanto, os vetores de suporte podem ser definidos como os exemplos da base de treinamento mais próximos do hiperplano. Em outras palavras, para encontrar o hiperplano, o SVM minimiza o limite superior do erro de generalização, que é obtido ao se maximizar a distância entre os pontos mais próximos em relação a cada uma das classes. A distância entre o hiperplano e os vetores de suporte define o tamanho da margem.

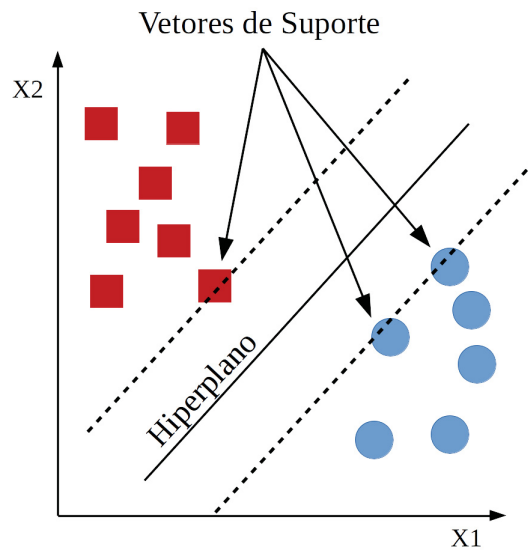


Figura 2.2: Hiperplano gerado pelo SVM.

O SVM faz uso de um mecanismo *Soft Margin* para lidar com conjuntos de treinamento que possuem *outliers* (exemplos na base de treinamento que não representam a classe em questão) que tornam os problemas não lineares. Esse mecanismo faz com que os *outliers* sejam descartados, fazendo uso do parâmetro C . Em outras palavras, quanto maior o valor de C , mais dados são descartados.

2.2 APRENDIZADO PROFUNDO

O desempenho de um sistema de classificação depende fortemente da escolha do conjunto de características (representação) utilizadas (Bengio et al., 2013). Por esse motivo, grande parte do esforço no desenvolvimento de aplicações em aprendizagem de máquina é destinado a melhorar as etapas de pré-processamento e transformações de dados, de modo a se obter um conjunto de características com alto poder de discriminação.

Em geral, essa representação de características é definida empiricamente por um especialista humano que, com conhecimento prévio em relação ao problema, julga quais características possuem maior poder de discriminação para determinada aplicação. Desse modo, é realizada a extração de tais características por meio de um determinado método.

Nesse contexto, o aprendizado automático da representação (sem a necessidade do ser humano) pode facilitar a extração de informações importantes na construção de sistemas de classificação, detecção ou reconhecimento. Entre as formas de aprendizagem, pode-se citar a *Representation Learning*, que consiste em um conjunto de métodos em que as representações necessárias para detecção ou classificação são descobertas automaticamente a partir de dados brutos (LeCun et al., 2015).

O problema central na *Representation Learning* é que pode ser muito difícil extrair características abstratas de alto nível a partir de dados brutos. O aprendizado profundo resolve esse problema introduzindo representações que são expressas em termos de outras representações mais simples (Goodfellow et al., 2016).

Como pode ser visualizado na Figura 2.3, na primeira camada de representação são extraídas características referentes à presença ou ausência de bordas em orientações e localizações específicas na imagem. Em seguida, cantos e contornos são detectados na segunda camada. A terceira camada é onde partes dos objetos são encontradas, localizando conjuntos específicos de

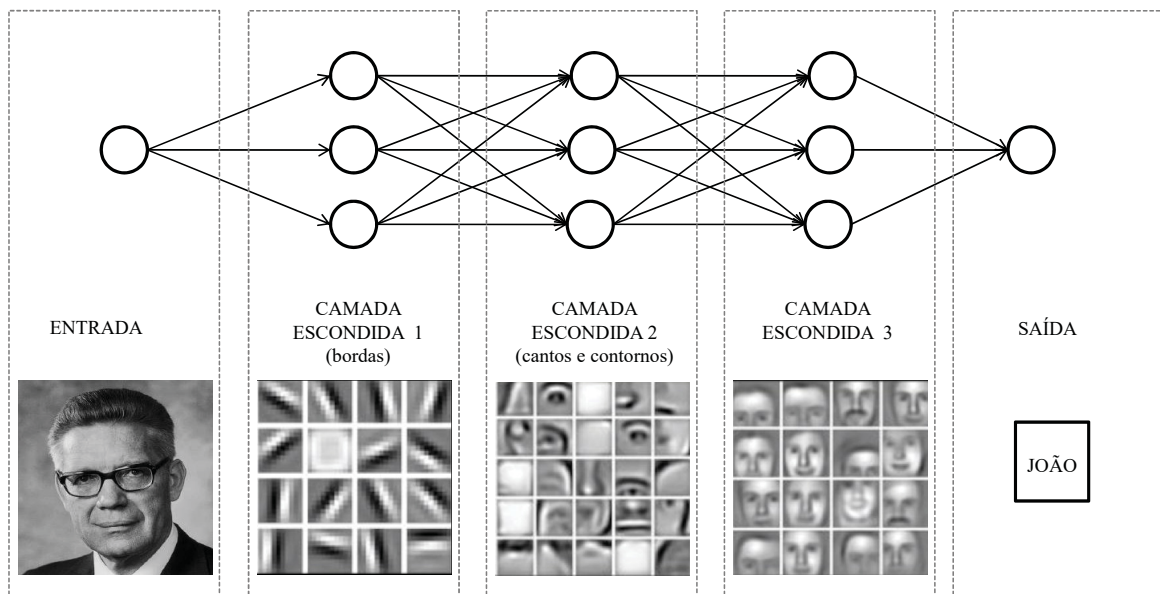


Figura 2.3: Ilustração de um modelo de aprendizado profundo.

contornos e cantos. Finalmente, as camadas subsequentes detectariam objetos específicos como combinações das partes encontradas na camada anterior (Goodfellow et al., 2016).

Segundo a observação feita por LeCun et al. (2015), uma das principais características da aprendizagem profunda é que as camadas de representações são aprendidas por meio de um procedimento de aprendizado com propósito geral e, portanto, requer pouca interferência humana.

Inicialmente, as abordagens de aprendizado profundo foram aplicadas, principalmente, no problema de reconhecimento de dígitos manuscritos, superando os resultados obtidos pelo *Support Vector Machine* (SVM). Posteriormente, o aprendizado profundo foi adotado em abordagens para reconhecimento de objetos em imagens naturais, podendo-se citar o avanço alcançado por (Krizhevsky et al., 2012b) no *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), reduzindo a taxa de erro do estado da arte de 26,2% para 15,3% (Bengio et al., 2013).

2.2.1 Redes Neurais Convolucionais

Um exemplo de abordagem baseada em Aprendizagem Profunda são as Redes Neurais Convolucionais - *Convolutional Neural Network* (CNN), que é um conceito introduzido por Fukushima e Miyake (1982) e popularizado por LeCun et al. (1998). Uma CNN é uma variação de uma Rede Perceptron Multicamada e consiste em camadas com diferentes funções. Essas redes possuem a capacidade de aprender representações com base em um conjunto de amostras de treinamento. As CNNs têm sido atualmente empregadas para melhorar os resultados do estado-da-arte em diversos problemas de reconhecimento e/ou classificação de imagens, tais como: abordagens biométricas (Lucio et al., 2018; Bezerra et al., 2018), sistemas de segurança e monitoramento, reconhecimento de placas de veículos (Laroca et al., 2018, 2019b), dentre outros (LeCun et al., 2015; Ahuja et al., 2017; Dumoulin e Visin, 2016; Krizhevsky et al., 2012b).

De modo geral, os exemplos são inicialmente aplicados nas camadas de entrada, conhecidas como camadas convolucionais (Vargas et al., 2016). Essas camadas são compostas por neurônios e cada neurônio é responsável por aplicar um filtro em uma área específica da imagem. Basicamente, um neurônio está conectado a um conjunto de pixels da camada anterior e para cada conexão é atribuído um peso. Os respectivos pesos de suas conexões produzem uma

saída que é passada para a próxima camada. Os pesos atribuídos às conexões de um neurônio podem ser interpretados como uma matriz representando o filtro de uma convolução de imagens no domínio espacial (*kernel*). Os pesos são compartilhados entre os neurônios de uma mesma camada, levando os filtros a aprender padrões que ocorrem na imagem.

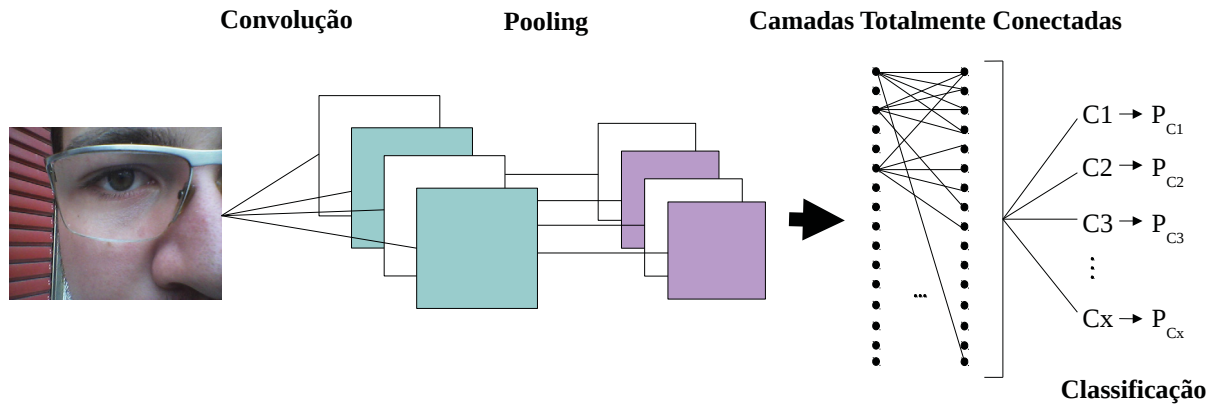


Figura 2.4: Exemplo das camadas de uma Rede Neural Convolucional.

Nesse contexto, normalmente existem quatro tipos básicos de camadas em uma CNN: Camada de Convolução, Camada de não linearidade (*Rectified Linear Unit*), Camada de *Pooling* (Agregação) e camada de Classificação (*Fully Connected Layer*). A Figura 2.4 ilustra a organização dessas camadas.

Convolução - A primeira camada sempre consiste em uma camada do tipo *Convolutional*. Um filtro (também chamado de *kernel*) com um conjunto de pesos é deslizado pela imagem e o resultado dessa convolução é atualizado no mapa de características (*feature map*) (Dumoulin e Visin, 2016). Cada filtro pode ser visto como um descritor de características, tais como bordas, linhas, curvas, cores, dentre outras.

Rectified Linear Unit (ReLU) - Basicamente, o objetivo da camada ReLU é introduzir não linearidade na rede, uma vez que a maioria dos dados no mundo real são não lineares (e a operação de convolução é linear). Essa camada realiza essa função por meio de um operador, que basicamente substitui todos os valores negativos do mapa de ativação por zero. Além disso, a ReLU resolve o problema do desaparecimento do gradiente (Vanish Gradient).

Pooling (Agregação) - Essa camada funciona de maneira semelhante à convolução, fazendo uso de um *kernel* que aplica um determinado cálculo nos valores de cada mapa de características. O objetivo consiste em reduzir a dimensionalidade de cada mapa de características, tentando conservar as informações mais relevantes. Consequentemente, essa técnica reduz a quantidade de parâmetros a serem aprendidos na rede, contribuindo para o controle de *overfitting*. A função aplicada pelo *kernel* pode ser implementada de diversas maneiras. Funções de *Máximo*, *Média*, *Soma* são alguns exemplos de possíveis funções a serem utilizadas na camada de *Pooling*.

Classificação (Camadas Totalmente Conectadas) - Essa camada consiste em um Perceptron Multicamadas, com função de ativação *softmax* na camada de saída, garantindo a estimação de probabilidades a posteriori.

Deste modo, a saída gerada por meio das camadas convolução+ReLU+Pooling representam características de alto nível que servem como entrada do Perceptron Multicamadas. Além disso, essa representação pode ser utilizada com qualquer outro classificador.

2.3 YOLO - DETECÇÃO DE OBJETOS COM ABORDAGEM DE APRENDIZADO PROFUNDO

No reconhecimento de padrões, os *pipelines* de detecção de objetos geralmente começam extraindo características das imagens de entrada em uma janela deslizante ou em algum subconjunto de regiões na imagem. Em seguida, os classificadores são usados para identificar objetos no conjunto de características (Redmon et al., 2016a).

Outra possibilidade é fazer uso de uma arquitetura de aprendizado profundo (VGGNet (Simonyan e Zisserman, 2014) ou GoogLeNet (Szegedy et al., 2015), por exemplo) originalmente utilizada para classificação de imagens e transformá-la em um detector de objetos. Nesse caso, seria passada uma janela deslizante pela imagem. Usar uma janela deslizante fornece centenas ou milhares de predições possíveis para essa imagem e serão consideradas as que o classificador reportar maior probabilidade. Essa abordagem funciona, entretanto será muito lenta, já que é necessário executar o classificador inúmeras vezes.

Uma abordagem mais eficiente consiste em, primeiramente, prever quais partes da imagem contêm informações de maior interesse — as chamadas Propostas de Região — e, posteriormente, executar o classificador apenas nessas regiões. Assim, o classificador precisará desempenhar um trabalho menor do que deslizar uma janela sobre toda a imagem inúmeras vezes (Girshick, 2015; Ren et al., 2015).

Ao contrário de técnicas baseadas em proposta de região ou janela deslizante, a Darknet YOLO (Redmon et al., 2016a) consiste em uma abordagem que reformula a detecção de objetos como um único problema de regressão, partindo dos pixels da imagem para coordenadas de *bounding boxes* e probabilidades da classe. Assim, a YOLO analisa globalmente a imagem no processo de detecção. Como o seu próprio nome (*You Only Look Once*) indica, a rede analisa a imagem apenas uma vez, porém de uma maneira inteligente (Redmon et al., 2016a).

De modo geral, a imagem é dividida em células que são responsáveis pela definição de *bounding boxes* (Figura 2.5(a)). As *bounding boxes* envolvem possíveis objetos a serem detectados. Por meio da definição de valores de confiança (gerados para cada *bounding box*) e de probabilidades classes para cada célula, mede-se a probabilidade de uma determinada região conter um tipo específico de objeto (Redmon et al., 2016a).

Por exemplo, a espessa *bounding box* amarela, ilustrada na Figura 2.5(b), apresentaria grande probabilidade de que exista um objeto da classe cachorro (caso a rede tenha sido treinada para isso) em seus limites.

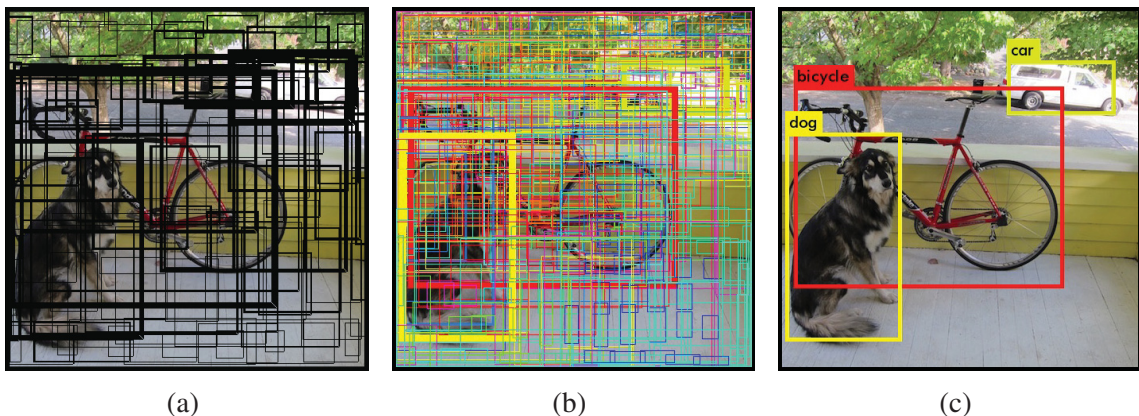


Figura 2.5: Detecção de objetos usando YOLO. Fonte: (Redmon, 2016b)

Um exemplo de como ficaria a detecção de objetos final pode ser visualizado na Figura 2.5(c). É importante ressaltar que, para realizar esse processo de detecção, a rede neural analisa a imagem apenas uma vez, sendo por esse motivo que a YOLO detecta objetos com maior velocidade. Assim como outras CNNs, A YOLO é composta de três camadas de operações principais para detecção de objetos, que são: *Convolução*, *Max Pooling* e *Classificação*, que ocorre por meio de camadas totalmente conectadas.

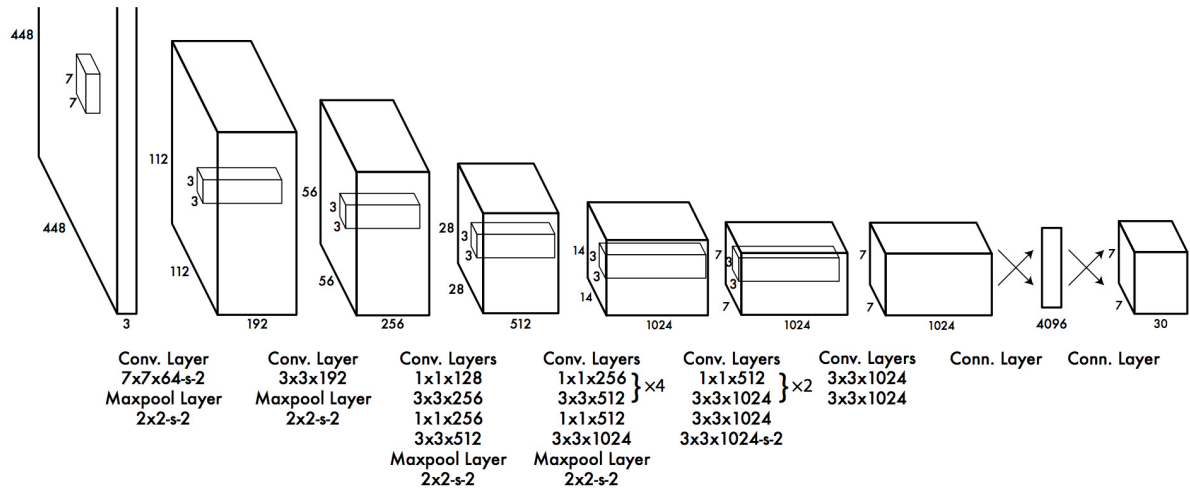


Figura 2.6: Arquitetura da YOLO. Possui 24 camadas convolucionais seguidas de 2 camadas totalmente conectadas. Camadas convolucionais alternadas de 1×1 reduzem o espaço de características das camadas anteriores. As camadas convolucionais são pré-treinadas na metade da resolução (224×224) e, em seguida, no dobro da resolução para detecção. Imagem reproduzida de (Redmon et al., 2016a).

Como pode ser visualizado na Figura 2.6, a YOLO tem 24 camadas convolucionais seguidas de 2 camadas totalmente conectadas. As primeiras 20 camadas convolucionais seguidas por uma camada de *pooling* e uma camada totalmente conectada foram usadas para pré-treinamento da tarefa de classificação na base de dados ImageNet. Em seguida, quatro camadas convolucionais e duas camadas totalmente conectadas foram adicionadas para detecção. Uma função de ativação linear é usada para a camada final e todas as outras camadas usam *Leaky ReLU*. Além disso, a resolução de entrada da rede foi aumentada de 224×224 para 448×448 , pois a detecção geralmente requer informações refinadas. Por fim, existe uma camada de *dropout* com taxa de 0,5 após a primeira camada conectada para evitar a co-adaptação entre as camadas (Redmon et al., 2016a).

Há também uma variação da versão original da YOLO, chamada de Fast YOLO (ou Tiny-YOLO). Em comparação com a versão original da YOLO, todos os parâmetros de treinamento e teste são os mesmos. No entanto, a Fast-YOLO utiliza um número menor de camadas convolucionais (9 em vez de 24) e menos filtros nessas camadas.

Apesar da Fast-YOLO ter uma arquitetura consideravelmente menor, ela ainda é capaz de detectar objetos com bastante precisão e mais rapidamente. Na Figura 2.7, são mostradas as detecções obtidas na mesma imagem com Fast-YOLO e YOLO, utilizando parâmetros padrão.

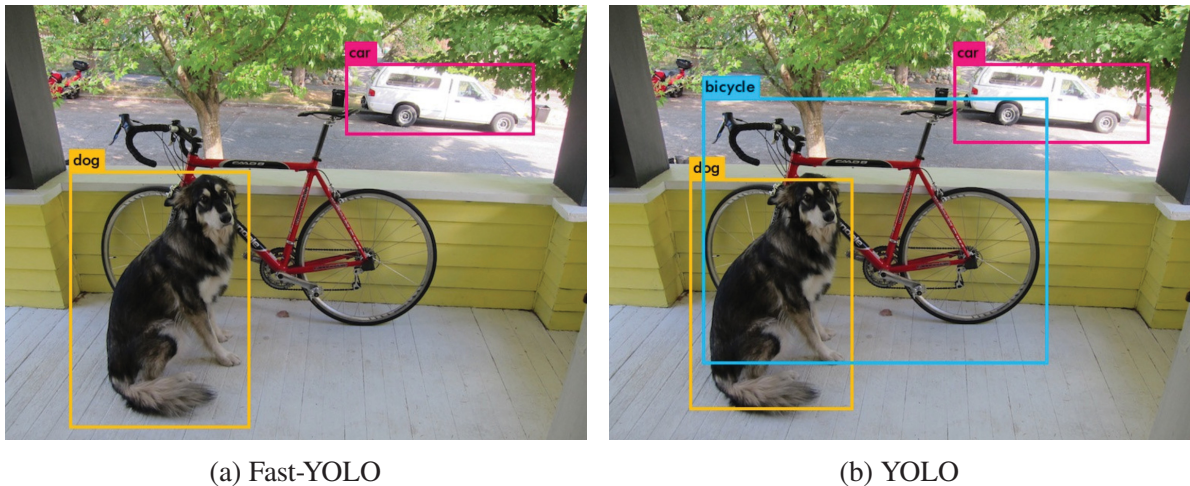


Figura 2.7: Exemplo de detecção de objetos com (a) Fast-YOLO e (b) YOLO. Imagem replicada de <https://pjreddie.com/darknet/yolo>.

Apesar dos resultados promissores alcançados, a YOLO tem algumas limitações. A primeira limitação observada é que cada célula da grade prediz, por padrão, apenas duas *bounding boxes* e só pode ter uma classe. Isso limita a detecção em regiões da imagem onde objetos estejam muito próximos, como ilustrado na Figura 2.8. Este não é um grande problema para a detecção de íris, visto que, considerando bases de dados da região periocular, não lidamos com alvos (íris) tão próximos (mais detalhes sobre as bases de imagens utilizadas são apresentados no Capítulo 4).



Figura 2.8: Ilustração de uma das limitações da YOLO, em que a detecção pode perder objetos muito próximos. Em (a), há 9 indivíduos no canto inferior esquerdo, porém apenas 5 foram detectados, como pode ser visto em (b). Imagem replicada de (Redmon et al., 2016a).

Outra observação a ser feita é que os erros em *bounding boxes* pequenas e grandes são tratados igualmente na função de perda. No entanto, um pequeno erro em uma *bounding box* grande pode não ser significativo, mas um pequeno erro em uma *bounding box* pequena tem um efeito muito maior na métrica IoU (Redmon et al., 2016a; Hui, 2018).

2.3.1 YOLOv2

Como reportado por Redmon e Farhadi (2017), a YOLO apresenta maiores erros de localização e o *recall* é menor quando comparado a alguns sistemas de detecção propostos posteriormente. A YOLOv2 (Redmon e Farhadi, 2017) é a segunda versão da YOLO com o objetivo de melhorar significativamente sua precisão e torná-la mais rápida.

Um novo modelo de classificação, chamado *Darknet-19*, é usado como base da YOLOv2. Tal modelo possui 19 camadas convolucionais e 5 camadas de *max-pooling*. Uma melhoria de mais de 2% em mAP foi alcançada adicionando *batch normalization* em todas as camadas convolucionais. Dessa forma, a camada de *dropout* foi removida sem gerar *overfitting*. Além disso, foi alcançado um aumento de quase 4% do mAP através da classificação em alta resolução, ou seja, após o treinamento da rede de classificação em imagens de 224×224 pixels, a rede foi ajustada na resolução de 448×448 no ImageNet (Redmon e Farhadi, 2017).

As camadas totalmente conectadas da YOLO foram removidas e a YOLOv2 usa *anchor boxes* para definir as *bounding boxes*. Assim, uma classe e a probabilidade de que a região contenha algum objeto são previstas para cada *anchor box* (Redmon e Farhadi, 2017). Para exemplificar, considere que 5 *anchor boxes* com proporções específicas sejam criadas, como ilustrado na Figura 2.9. Em vez de prever 5 *bounding boxes*, a YOLOv2 prevê compensações para cada uma dessas *anchor boxes*. Desse modo, a rede apenas ajusta o tamanho da âncora mais próxima ao tamanho do objeto. Essa característica, de acordo com (Redmon e Farhadi, 2017), simplifica o problema e facilita o aprendizado da rede. (Redmon e Farhadi, 2017). A YOLOv2 executa o algoritmo *k-means* nas caixas delimitadoras do conjunto de treinamento para encontrar automaticamente boas *anchor boxes*.

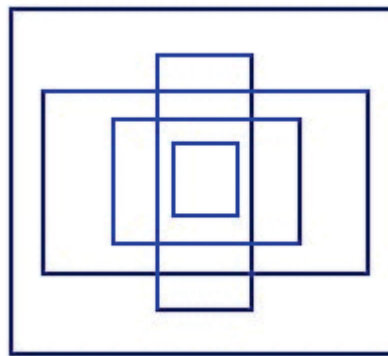


Figura 2.9: Exemplo de *anchor boxes*.

Outro conceito introduzido na YOLOv2 é o treinamento em várias escalas. A YOLO original usa uma resolução de entrada fixa de 448×448 pixels. Já na YOLOv2, a cada 10 lotes, a rede escolhe aleatoriamente um novo tamanho de dimensão da imagem entre 320×320 e 608×608 pixels (valores padrão). Essa abordagem força a rede a aprender uma significativa variedade de dimensões de entrada, aumentando o mAP em 1,4% (Redmon e Farhadi, 2017).

3 TRABALHOS RELACIONADOS

Encontram-se na literatura relevantes pesquisas e métodos computacionais que realizam a localização da íris. Este capítulo apresenta alguns desses trabalhos relacionados aos objetivos da nossa proposta. Na Seção 3.1, os trabalhos relacionados à localização de íris são apresentados. Na Seção 3.2, é realizada uma análise comparativa sobre tais trabalhos.

3.1 TRABALHOS RELACIONADOS A LOCALIZAÇÃO DE ÍRIS

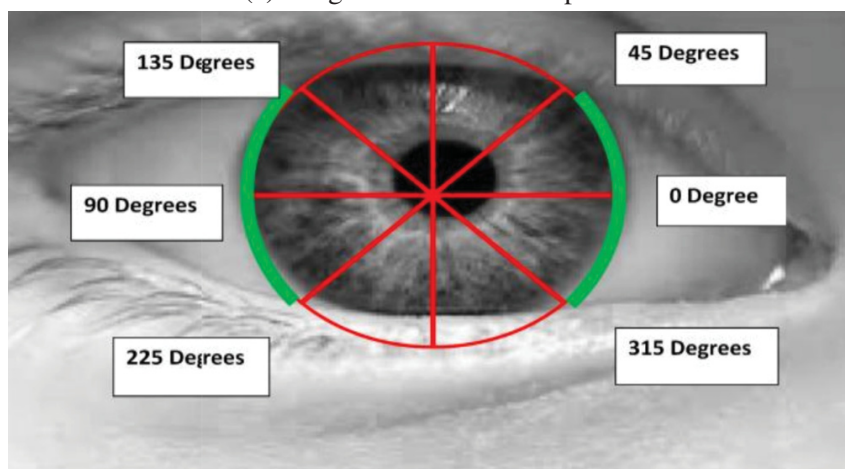
Um dos pioneiros e talvez mais conhecido método de localização de íris foi proposto por Daugman (Daugman, 1993). O método propõe um operador *integro-differential* para identificar as bordas circulares presentes na imagem. O primeiro passo consiste na utilização de um filtro gaussiano para suavizar a imagem, de modo a remover ruídos. Com a imagem suavizada, o operador busca a borda circular onde há a maior diferença dos valores de pixel, variando o raio r e o centro (x, y) do contorno circular. O operador é aplicado iterativamente com a quantidade de suavização progressivamente reduzida, a fim de buscar uma melhor precisão na localização.

A convolução é aplicada na imagem inteira e ocorre, basicamente, da seguinte maneira: Em cada iteração, em raio crescente, é realizada a soma normalizada de todos os valores de pixel circunferenciais. Em cada nível de raio aumentado, a diferença entre a soma normalizada dos pixels no círculo e a soma normalizada dos pixels pertencentes ao círculo adjacente é anotada. A circunferência definida como a localização da íris será aquela que apresentar a maior diferença em relação ao contorno adjacente.

Esse método normalmente falha ao ser utilizado em imagens que apresentam oclusão, como ilustrado na Figura 3.1(a). Nessas imagens, a íris é coberta total ou parcialmente pela pálpebra, podendo fazer com que seja detectada uma região de alto contraste. Para resolver esse problema, Daugman (2004) apresenta uma melhoria em sua abordagem para localizar a íris que consiste em considerar apenas os pixels pertencentes às porções laterais da circunferência, correspondentes aos ângulos de 0° à 45° , 135° à 225° e 315° à 360° , como ilustrado na Figura 3.1(b).



(a) Imagem com oclusão superior



(b) Porções da circunferência a serem consideradas com a melhoria do método de Daugman

Figura 3.1: Proposta de Daugman para mitigar o efeito de oclusões superiores e inferiores. Fonte: (Oad e Ahmad, 2012)

Outro método pioneiro para o problema de localização de íris foi proposto por (Wildes, 1997). A primeira etapa do método consiste em uma etapa de pré-processamento, em que, inicialmente, a imagem é reduzida por média espacial a fim de diminuir a demanda computacional para o restante do método. A próxima etapa do pré-processamento consiste na aplicação de um filtro gaussiano passa-baixa, de modo a suavizar os ruídos de alta frequência presentes na imagem.

Como a média espacial introduz redundância no domínio espacial, a imagem filtrada é subamostrada sem nenhuma perda adicional de informação. A imagem subamostrada serve como base para o processamento subsequente, com a vantagem de que suas dimensões menores e resolução menor levam a uma menor demanda computacional em comparação com a imagem de entrada original, em tamanho real.

O próximo passo consiste em aplicar o filtro de detecção de bordas *Canny*, que é um filtro baseado em gradiente. Esse filtro é configurado de modo a favorecer a detecção de bordas verticais. Os autores relatam que essa configuração é motivada pela observação de que, mesmo em imagens que apresentam pálpebras oclusivas, os limites laterais da íris normalmente são visíveis. Por fim, a transformada de *Hough* é aplicada para se definir a borda que melhor se encaixa em um círculo.

Os experimentos foram realizados com imagens coletadas de 50 voluntários (homens e mulheres) associados ao Centro de Pesquisa David Sarnoff. As íris desses indivíduos abrangiam a gama de cores comuns de íris: azul, marrom, verde, avelã e cinza. Na escuridão, o alcance era de azul claro a marrom muito escuro. Além disso, os autores relatam que os indivíduos que utilizavam óculos, retiraram o acessório durante a aquisição das imagens. Não foram dados mais detalhes sobre as imagens.

Rodríguez e Rubio (2005) apresentaram uma abordagem para localização de íris baseada no formato circular da pupila e em fundamentos de geometria analítica. Inicialmente, a pupila é isolada por meio de um método de binarização. O próximo objetivo consiste na definição de um ponto inicial PI , localizado dentro da região da pupila. Para isso, determina-se a linha L e coluna C com maior quantidade de pixels escuros contínuos (para se definir se um pixel é escuro ou não, é utilizado um limiar fixo). O ponto de intersecção entre a linha L e coluna C será definido como o ponto PI .

Considerando que, normalmente, há um significativo contraste entre a região da pupila e a região da íris, o próximo passo consiste em definir os pontos $P1$, $P2$ e $P3$ (ilustrados na Figura 3.2(a)) por meio de uma busca iterativa, iniciando a partir do ponto PI . A estratégia para encontrar os três pontos consiste em seguir três trajetórias cujos ângulos são 0° , 120° e 240° , em uma vizinhança 3×3 . Em cada iteração, é calculada a diferença entre o desvio padrão do ponto inicial e dos pontos da trajetória. O ponto cuja diferença for maior que um determinado limiar, será selecionado como ponto pertencente ao contorno da pupila. Esse processo é fundamentado no fato de que há pouca variação do desvio padrão dentro da pupila. Entretanto, o desvio padrão sofre um aumento abrupto, uma vez que a vizinhança 3×3 começa a receber pixels pertencentes à região da íris.

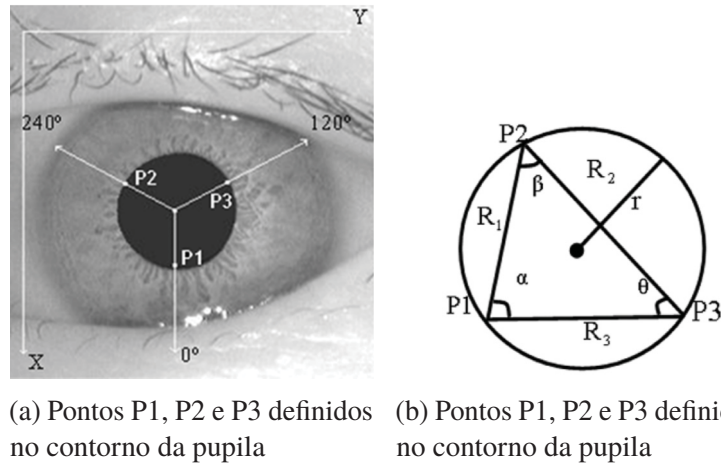


Figura 3.2: Pontos e vértices definidos no método de Rodríguez e Rubio, 2005. Fonte: (Rodríguez e Rubio, 2005)

Estes três pontos permitem definir os vértices de um triângulo, como pode ser visualizado na Figura 3.2(b). Assim, é possível determinar uma circunferência (localização da pupila) por meio de fundamentos de geometria analítica. A região da pupila é utilizada na localização da íris, em que é realizada uma busca por mudanças abruptas de gradiente por meio do método de Daugman. Os autores relatam que o método proposto não resulta em acurácia melhor do que o método proposto por Daugman. Entretanto, o algoritmo utilizado para localizar a pupila proporciona uma redução no custo computacional, em relação ao método de Daugman (Rodríguez e Rubio, 2005).

Alvarez e Garcia (2010) publicaram um método para localização de íris baseado em aproximação de gradiente por meio de operadores de *Ordered Weighted Averaging* (OWA) (Peláez

e Doña, 2006). Nesse método, primeiramente é aplicado um filtro de média, a fim de suavizar certos ruídos da imagem. O filtro de média é implementado com uma máscara de convolução 15×15 . Desta forma, busca-se eliminar os detalhes da imagem que podem causar segmentações errôneas da região íris. Além disso, o efeito negativo das reflexões especulares no processo de localização da íris é mitigado.

O próximo passo consiste em definir uma aproximação inicial do centro da pupila. Para isso, o primeiro passo consiste em aplicar o *threshold* de Otsu para binarizar a imagem. O contorno inicial da pupila será definido por meio das bordas do maior objeto presente na imagem binarizada e uma aproximação do centro da pupila será, portanto, o centro de tal objeto. A definição do centro da pupila permite a utilização de um método de localização de íris baseado na detecção de limites circulares por meio uma abordagem de análise de gradiente em pontos de interesse de arcos sucessivos. O operador *Quantified Majority Operator* (QMA), proposto por Peláez e Doña (2006), foi utilizado a fim de obter um valor representativo para cada arco. A localização da região da íris ocorre por meio da obtenção do arco com o melhor valor representativo. Os autores reportaram resultados similares aos resultados atingidos pelo método de Daugman. Entretanto, podem ser observadas melhorias no tempo de processamento.

O método proposto por Sundaram et al. (2011) busca localizar a íris por meio de detecção de pontos de borda e Transformada de Hough Circular. A abordagem pode ser dividida em cinco etapas principais:

- Primeiramente, é considerada apenas uma faixa horizontal (FH) da imagem original. Essa faixa começa na linha do meio e possui altura de um quarto da altura da imagem. A FH é definida com a suposição de que uma porção significativa da íris deve ser coberta pela FH.
- O próximo passo consiste em binarizar a região delimitada pela faixa definida na etapa anterior. Para isso, é calculado o histograma cumulativo (h) e o limiar th , calculado por: $th = \max_t \{h(t) \leq \frac{h(max)}{2}\}$, onde max é o nível de cinza máximo de FH. Com a imagem binarizada, é selecionado o maior elemento na imagem.
- Calcular a média (m) e desvio padrão (σ) dos pixels originais pertencentes à região do maior elemento. Sendo assim, na imagem original (inteira), são considerados apenas os pixels em que a intensidade está entre $[m - 2\sigma, m + 2\sigma]$. A região formada por tais pixels define a região de interesse.
- O detector de bordas Canny é aplicado na região de interesse definida no passo anterior. Como resultado, vários componentes de borda são obtidos. Destes, os menores componentes são removidos.
- Por fim, é aplicada a Transformada de Hough Circular neste conjunto de bordas reduzido e isso retornará a localização da íris.

Cui et al. (2012) apresentaram um método para localização de íris baseado em morfologia matemática. O primeiro passo do método consiste em uma etapa de pré-processamento. Inicialmente, a imagem original é transformada em escala de cinza e é aplicado um operador de realce. Na sequência, o método busca remover ruídos (cílios, manchas, imagens borradas, dentre outros) no entorno da íris e remover pontos fora da íris em que o nível de cinza é igual ao pico de cinza. Para isso, é utilizado um filtro passa baixa bidirecional gaussiano. O próximo passo consiste em obter uma localização inicial da pupila. Para isso, a imagem preprocessada

é binarizada. O limiar utilizado na binarização é calculado por meio de *Adaptive threshold* (Xiaorong e Yongjun, 2011).

A partir da imagem binarizada, operadores de morfologia matemática são aplicados, tais como erosão, dilatação, abertura e fechamento. Tais operadores são aplicados para remover interferências remanescentes, causadas pelos cílios. A imagem resultante consiste na localização aproximada da pupila. A localização precisa da pupila é realizada por meio do operador de detecção de correspondência circular proposto por Daugman.

O próximo passo consiste na localização dos limites externos da íris. Primeiramente, é considerado o centro da pupila, localizada anteriormente, como ponto de partida para se encontrar o centro da íris (que normalmente são muito próximos).

Primeiramente, tomando o centro da pupila como a base da coordenada, é definida uma linha reta S_2 na direção horizontal. Adicionalmente, são definidas outras duas retas horizontais, S_1 e S_3 . Tais retas consistem nos limites superior e inferior para a localização da íris. Para definir a posição de S_1 e S_3 , os autores relatam que definiram uma distância de 30 pixels (superior e inferior) em relação a reta S_2 .

Finalmente, a localização da região da íris é obtida fazendo uso da Transformada de Hough, juntamente com o Método dos Mínimos Quadrados (MMQ).

Zhang e Ma (2014) abordaram um método para localização de íris baseado em *level set*. O primeiro passo consiste em encontrar uma região de interesse. Essa região de interesse é encontrada por meio do método descrito por Li et al. (2012), baseado em *Histogram of Oriented Gradients*. O próximo passo consiste em determinar um contorno inicial para a pupila. Esse contorno inicial é definido da seguinte maneira:

- A região de interesse é dividida em blocos de tamanho fixo. Os autores relatam que utilizaram blocos de tamanho 20×20 pixels.
- O valor de cinza médio de cada bloco é calculado, de modo a utilizar o valor correspondente a menor média como limiar para a binarização da imagem.
- Operadores de morfologia matemática são aplicados (dilatação, erosão e preenchimento de buracos), para se obter uma borda preliminar da pupila.

Tendo um contorno preliminar da pupila, o próximo passo consiste em determinar o contorno preciso da pupila, utilizando um método de *level set* baseado em *momentum*. Por fim, o operador *integro-differential*, inicialmente utilizado por Daugman, é aplicado, de modo a localizar a íris.

No trabalho de Su et al. (2017), foi proposto um algoritmo para localização de íris baseado em *Iterative Searching*. Na primeira etapa, a imagem é convertida para escala de cinza e é binarizada, utilizando um limiar variável, calculado de acordo com o histograma de cada imagem. A segunda etapa consiste em determinar a região da pupila com base em propriedades regionais. Para isso, a imagem binária é convertida em uma matriz de rótulos. É definida como a área da íris o maior objeto nessa matriz de rótulos. Por fim, o contorno da pupila é definido por meio do filtro Sobel. Os valores de centro e raio da pupila são definidos por meio do contorno detectado.

Com a conclusão da localização da pupila, é aplicado o método de localização da íris. Tal método é baseado na diferença de níveis de cinza observado entre a íris e a esclera. O método busca encontrar os pontos laterais da íris (pontos A e D ilustrados na Figura 3.3) por meio de uma busca iterativa, que ocorre da seguinte maneira: Partindo do ponto C para a direita, em cada pixel, é aplicado o cálculo descrito na Equação 3.1, em que é quantificada a diferença de

intensidade em relação aos 15 pixels posteriores. O pixel que apresentar maior diferença será definido como o ponto D .

$$sum = \sum_{x=1}^{15} f(i+x, y) - \sum_{x=1}^{15} f(i-x, y) \quad (3.1)$$

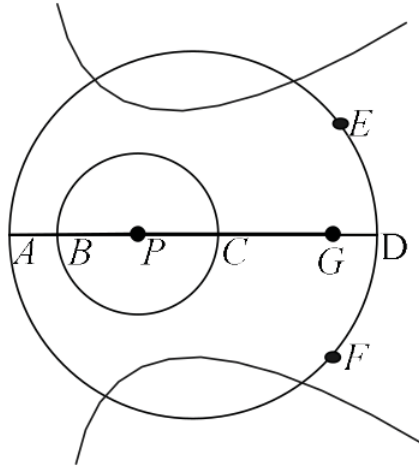


Figura 3.3: Pontos a serem detectados no método de Su et al. (2017). Fonte: (Su et al. 2017).

De modo semelhante é calculado o ponto A , aplicando uma busca iterativa partindo do ponto C , para a esquerda. Por meio da Equação 3.2, é calculado o pixel em que apresenta maior diferença em relação à vizinhança.

$$sum = \sum_{x=1}^{15} f(i-x, y) - \sum_{x=1}^{15} f(i+x, y) \quad (3.2)$$

Por fim, o objetivo é localizar os pontos E e F . Para isso, o método adotado pelos autores consiste em calcular um ponto G , conforme demonstrado na Equação 3.3.

$$dist(A, G) = 0,95 \times dist(A, D) \quad (3.3)$$

Com o ponto G calculado, são calculados os pontos E acima e F por meio do mesmo método utilizado para definir os pontos A e D . Com todos os pontos calculados, é possível obter a localização da íris.

3.2 ANÁLISE DOS TRABALHOS RELACIONADOS À LOCALIZAÇÃO DE ÍRIS

Nessa seção, faremos uma análise das abordagens de localização de íris apresentadas anteriormente. Em tal análise, serão discutidas as principais vantagens e limitações dos algoritmos utilizados por essas abordagens. Na Tabela 3.1, são apresentadas, de forma sumarizada as principais observações acerca dos métodos encontrados na literatura.

Por meio das considerações apresentadas na Tabela 3.1, observam-se diversos esforços para tratar e mitigar os desafios enfrentados pelos métodos de localização de íris, o que resultou em diversas melhorias e sugestões apresentadas nos últimos anos. Entretanto, ainda observam-se diferentes limitações nas abordagens de localização da íris encontradas na literatura. Dentre tais limitações, destacam-se, por exemplo, a dependência por imagens em que não há oclusão lateral

Método	Base de Dados	Resultados (%)	Vantagens	Limitações
(Daugman, 2004)	Não publicada	-	- Considerar apenas as porções laterais da circunferência possibilita mitigar o efeito de oclusões que frequentemente aparecem na parte superior ou inferior da íris.	- Dependência por imagens que apresentam alto contraste entre íris e o restante da imagem. - Reflexos fazem com que sejam formadas regiões de alto contraste, fazendo com o método as identifique como íris. - O método provavelmente irá falhar em imagens que a as regiões laterais da íris não são visíveis.
(Wildes, 1997)	Não publicada	-	- Em uma etapa de préprocessamento, a imagem de entrada é reduzida e, consequentemente, a demanda computacional no restante do método diminui. - De acordo com experimentos realizados, encontrados na literatura, a utilização da Transformada e Hough apresenta uma melhor acurácia em relação ao operador <i>integro-differential</i> de Daugman (Patil, 2014; Zhou et al., 2013).	- A detecção de contornos é configurada de modo a favorecer bordas verticais. Entretanto, há muitos casos em que a oclusão não é apenas superior ou inferior, e sim lateral; - Considera que as regiões para os pontos P1, P2 e P3 sempre serão "visíveis". - Dependência por imagens que apresentam alto contraste entre íris e o restante da imagem.
(Rodríguez e Rubio, 2005)	CASIA Iris Image	Acurácia 98,00	- Os autores relatam que algoritmo utilizado para localizar a pupila proporciona uma redução no custo computacional para se localizar a íris, em comparação com método de Daugman (Rodríguez e Rubio, 2005).	- Aplicar, inicialmente, um método de binarização (com a finalidade de isolar a pupila) pode ser uma limitação em bases com imagens obtidas de distâncias maiores ou bases construídas em ambientes não controlados, visto que tais imagens frequentemente apresentam diversas estruturas conectadas na pupila e de intensidade de pixel semelhante (óculos, cabelo, parte do nariz, etc).
(Alvarez e Garcia, 2010)	CASIA-Iris V3	Acurácia 98,00	- O efeito negativo das reflexões especulares é mitigado por meio de uma etapa inicial de préprocessamento; - Os autores reportaram resultados similares aos resultados atingidos pelo método de Daugman. Entretanto, podem ser observadas melhorias no tempo de processamento.	- O operador QMA, utilizado para localizar a íris, é baseado na detecção de círculos. Entretanto, na maioria das bases de imagens VIS (MICHE-I, UBIRIS, MobBio), a íris não corresponde a uma região circular, devido a oclusões, cabelo, óculos, etc.
(Sundaram et al., 2011)	UBIRIS	Acurácia 98,43	- Define uma região de interesse, o que pode eliminar a interferência de demais estruturas no processo de localização da íris.	- Em imagens obtidas de distâncias maiores (MICHE-I, por exemplo), não há uma garantia de que a região região de interesse HB irá cobrir a íris (ou parte dela). Nesses casos, a localização será afetada negativamente.
(Cui et al., 2012)	CASIA-iris-Twins	Acurácia 98,84	- Executa uma etapa de préprocessamento, buscando remover cílios e reflexos. A correta remoção (ou suavização) de tais ruídos pode melhorar a precisão da posterior localização da íris.	- A utilização do algoritmo de binarização e de operadores de morfologia matemática torna o método específico para a base utilizada nos experimentos, em consequência dos parâmetros (fixos) utilizados.
(Zhang e Ma, 2014)	CASIA Iris V2	Acurácia 98,53	- Inicialmente, define uma região de interesse, o que pode eliminar a interferência de demais estruturas no processo de localização da íris. - Estima uma localização preliminar para a pupila, o que facilita na detecção precisa das bordas da pupila e, consequentemente, viabilizando uma melhor localização da íris.	- Na etapa final de localização de íris, faz uso do método de Daugman e, portanto, depende de um significativo contraste entre a região da íris e sua vizinhança. - Também depende de imagens em que as regiões laterais da íris sejam visíveis.
(Su et al., 2017)	CASIA V3 Interval	Acurácia 98,08	- Os autores reportam que o método para detecção do contorno interno da íris apresenta maior acurácia em comparação com as demais abordagens existentes. Em abordagens que localizam o contorno interno da íris como etapa inicial, a precisão da detecção de tal contorno pode afetar diretamente a localização final da íris. - Por meio de experimentos realizados pelos autores, foi observado que o método demanda um menor custo computacional em comparação com o método de Daugman e o método de Wildes.	- Em imagens com ruídos escuros conectados na pupila (cabelo ou óculos, por exemplo), a localização da pupila provavelmente não será correta, comprometendo a posterior localização da íris. - Dependência por imagens que apresentam alto contraste entre íris e o restante da imagem. - Considera que sempre haverá limites visíveis da íris nas regiões dos pontos <i>D</i> , <i>E</i> e <i>F</i> (íris sem oclusão lateral).

Tabela 3.1: Análise sumarizada dos trabalhos relacionados.

da íris, dependência por imagens em que a íris apresenta um significativo contraste em relação à sua vizinhança, experimentos realizados em uma única base de dados ou em apenas um domínio (VIS ou NIR).

Além disso, observou-se que todos os trabalhos encontrados na literatura para localização de íris quantificam seus resultados utilizando apenas a métrica *Accuracy*. Considerando o problema de localização de íris, a métrica acurácia não possibilita uma avaliação realista dos métodos propostos, visto que imagens da região periocular representam, em sua grande maioria, dados desbalanceados, já que a íris (classe positiva) é uma pequena região em relação ao restante da imagem (classe negativa). Como visto anteriormente, não é indicada a utilização da métrica *Acurácia* em exemplos desbalanceados.

Adicionalmente, não foi encontrada uma abordagem de localização de íris que tenha sido experimentada em imagens tomadas de distâncias maiores, tais como imagens encontradas na base MICHE ou BERC (agregando significativamente a presença de ruídos, tais como óculos, cabelo, parte do nariz, etc).

4 MATERIAIS E MÉTODOS

Nesse capítulo, são abordados os materiais utilizados na realização desse trabalho, bem como os métodos propostos para localização da íris em imagens da região periocular.

Primeiramente, na Seção 4.1, são apresentadas e discutidas as bases de dados utilizadas nos experimentos. Na Seção 4.2, abordaremos as métricas de avaliação que normalmente são empregadas em detecção de objetos e que foram aplicadas nesse trabalho. Por fim, na Seção 4.3, são apresentadas as abordagens propostas nesse trabalho.

4.1 BASES DE DADOS

Atualmente, existem várias bases de imagens da região periocular, criadas de diferentes maneiras e com o uso de diferentes equipamentos, sendo utilizadas para diversos propósitos. De modo geral, as bases de imagens da região periocular podem ser classificadas em dois tipos: cooperativas e não-cooperativas. As bases de dados cooperativas possuem imagens capturadas em ambientes com condições controladas, tais como iluminação, distância do equipamento para a íris, foco, dentre outros. Bases de dados não-cooperativas são construídas com imagens obtidas em ambientes não controlados, apresentando problemas de foco, oclusão, reflexos, variações de ângulos, etc. Além disso, dependendo do equipamento, as imagens podem ser capturadas em comprimento de onda no espectro visível (VIS) ou infravermelho próximo — *near-infrared* (NIR).

4.1.1 Bases de Imagens NIR

Imagens da região periocular capturadas em espectro infravermelho próximo são, geralmente, utilizadas para estudar características presentes na íris (Phillips et al., 2007, 2008; Daugman, 2004; Bowyer et al., 2008). Adicionalmente, esse tipo de imagem é também utilizado em estudos de diferentes aplicações, tais como metodologias para a criação de íris sintéticas (Shah e Ross, 2006; Zuo et al., 2007), vulnerabilidades no reconhecimento de íris, detecção de vivacidade (*liveness*) em imagens da íris (Ruiz-Albacete et al., 2008; Czajka, 2013; Gupta et al., 2014; Kohli et al., 2016), impacto do uso de lentes de contato em métodos de reconhecimento de íris (Baker et al., 2010; Kohli et al., 2013; Doyle et al., 2013; Doyle e Bowyer, 2015), avaliação do impacto causado pelo envelhecimento do indivíduo em sistemas biométricos (Fenker e Bowyer, 2012; Baker et al., 2013), influência do consumo de álcool no reconhecimento de íris (Arora et al., 2012), reconhecimento de gênero através da íris (Tapia et al., 2016), dentre outros.

Nos experimentos realizados nesse trabalho, utilizamos seis bases de imagens NIR, sendo elas: IIIT-Delhi Contact Lens Iris (IIIT-D CLI) (Kohli et al., 2013), CASIA Iris Image Database Version 2.0 (CASIA-IrisV2, 2004), Notre Dame Contact Lens Detection 2015 (NDCLD15) (Doyle e Bowyer, 2015), Notre Dame Cosmetic Contact Lenses (NDCCL) (Doyle e Bowyer, 2014), CASIA-IrisV3 Interval (CASIA, 2010) e BERC Mobile-iris (Kim et al., 2016). Na sequência, apresentamos as informações de tais bases de dados, bem como exemplos de imagens de cada uma delas. Adicionalmente, a Tabela 4.1 apresenta, de forma resumida, as principais características das bases de imagens NIR utilizadas nesse trabalho.

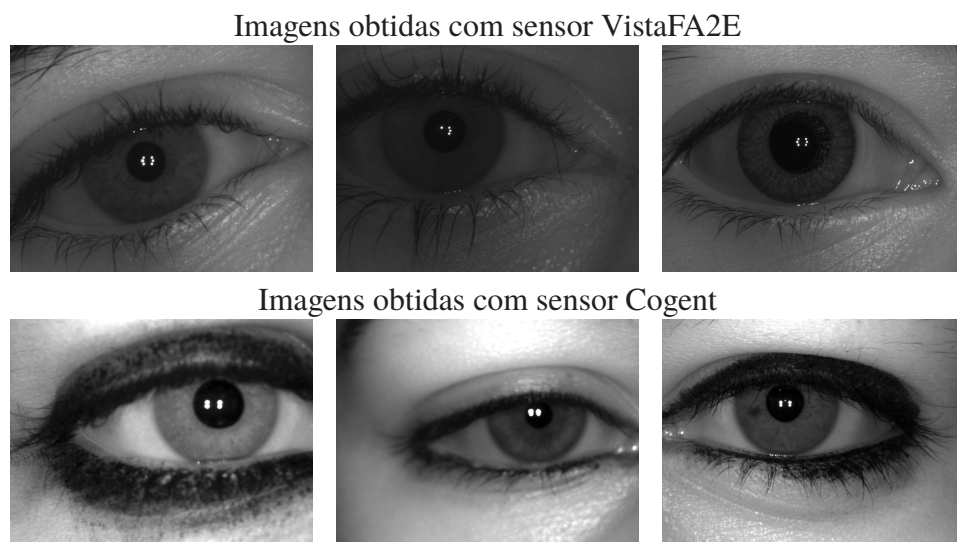
IIIT-Delhi Contact Lens Iris: A base de imagens IIIT-D CLI é composta por 6570 imagens da região periocular, de 101 indivíduos. São utilizadas três classes de imagens para a composição da base: imagens da região periocular de indivíduos que não utilizam lentes de

Tabela 4.1: Principais informações das bases de imagens NIR

Base de Dados	Ano	Resumo	Ambiente Controlado	Nº de Imagens
IIIT-D CLI	2013	Analisar o efeito do uso de lentes de contato em reconhecimento de íris	Sim	6570
CASIA-IrisV3 Interval	2010	Estudar detalhes das características de textura nas imagens de íris	Sim	2639
NDCCL	2014	Detectar e identificar tipos de lentes de contato	Sim	4200
NDCLD15	2015	Classificação de imagens da íris de acordo com o tipo da lente de contato usada	Sim	7300
BERC	2016	Reconhecimento de íris usando imagens de dispositivos móveis	Sim	500
CASIA-IrisV2	2015	Estudo de característica da íris com imagens capturadas em NIR	Sim	2400

contato, indivíduos que utilizam lentes de contato transparentes e indivíduos que utilizam lentes de contato com textura colorida.

A fim de se estudar o efeito que o dispositivo de aquisição tem sobre a imagem, as imagens foram capturadas utilizando dois sensores diferentes: Cogent e VistaFA2E. Foram capturadas pelo menos 5 imagens de cada classe de íris em cada categoria de lente para os dois sensores (Kohli et al., 2013). Todas as imagens dessa base possuem resolução de 640×480 pixels e foram manualmente rotuladas. Na Figura 4.1, exemplos de imagens da base IIIT-D CLI podem ser visualizados.

Figura 4.1: Exemplos de imagens encontradas na base de dados *IIIT-Delhi Contact Lens Iris*

CASIA-IrisV3 Interval - A base de dados CASIA-IrisV3 Interval consiste em 2639 imagens da região periocular, obtidas em duas seções. Um exemplo das imagens pertencentes a essa base pode ser visualizado na Figura 4.2.

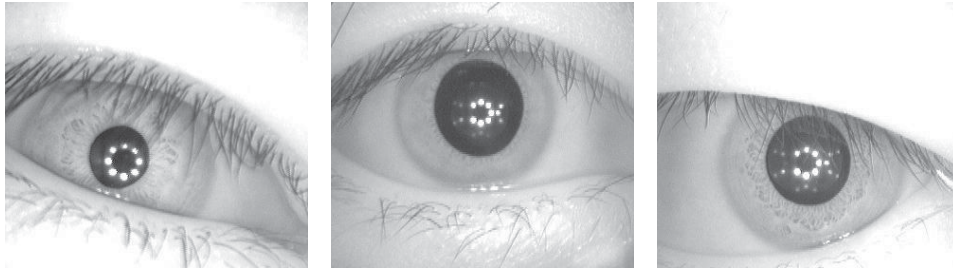


Figura 4.2: Exemplos de imagens encontradas na base de dados *CASIA-IrisV3 - Interval*

As imagens possuem resolução de 320×280 pixels e foram capturadas por uma câmera desenvolvida pelos próprios criadores da base, tendo um conjunto circular de *leds* infravermelho próximo. Os autores relatam que as configurações da câmera permitem que as imagens sejam utilizadas para se estudar detalhes das características de textura nas imagens da íris (CASIA, 2010).

Notre Dame Cosmetic Contact Lenses - As imagens da base de dados NDCCL possuem resolução de 640×480 pixels e foram obtidas com luz de infravermelho próxima. Foram utilizadas duas câmeras para se obter as imagens: IrisGuard AD100 e IrisAccess LG4000, compondo dois subconjuntos. Exemplos de imagens de ambos os subconjuntos podem ser visualizados na Figura 4.3

Imagens tomadas com sensor IrisGuard AD100



Imagens tomadas com sensor IrisAccess LG400

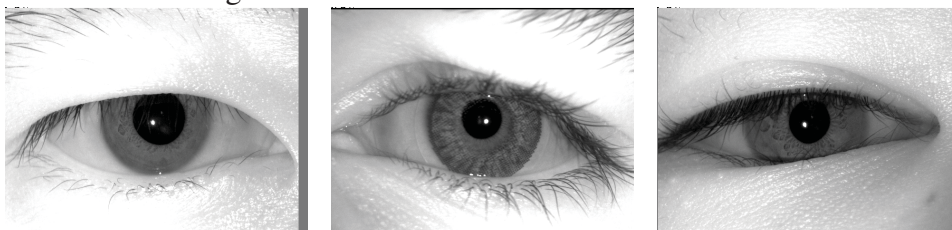


Figura 4.3: Exemplos de imagens encontradas na base de dados *Notre Dame Cosmetic Contact Lenses*

O subconjunto IrisAccess LG4000 é formado por 4200 imagens e o subconjunto IrisGuard AD100 possui 900 imagens (Doyle et al., 2013; Yadav et al., 2014). Assim como a base de dados IIIT-D CLI, a base NDCCL contém imagens divididas em três classes: indivíduos que não estão utilizando lentes de contato, indivíduos utilizando lentes de contato transparentes e indivíduos utilizando lentes de contato com textura colorida.

Notre Dame Contact Lens Detection 2015 - A base de dados NDCLD15 consiste em 7300 imagens da região periocular, sendo 6000 imagens compondo o conjunto de treinamento e

1300 destinadas para avaliação. As imagens dessa base de dados possuem resolução de 640×480 pixels e foram obtidas utilizando dois sensores: IrisAccess LG4000 e Iris-Guard AD100. Todas as imagens foram capturadas em um laboratório sem janelas, sob consistentes condições de iluminação, sendo criada com o propósito de estudar a classificação de lentes de contato em imagens da região periocular (Doyle e Bowyer, 2015). Portanto, assim como outras bases de dados criadas com o mesmo propósito, a NDCLD15 contém imagens divididas em três classes: indivíduos que não estão utilizando lentes de contato, indivíduos utilizando lentes de contato transparente e indivíduos utilizando lentes de contato com textura colorida. Exemplo de imagens encontradas nessa base podem ser visualizados na Figura 4.4.

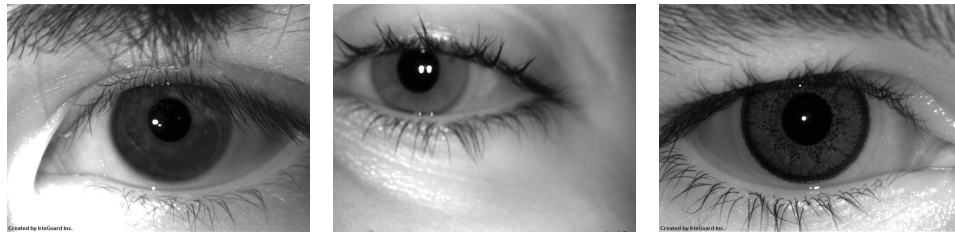


Figura 4.4: Exemplos de imagens encontradas na base de dados *Notre Dame Contact Lens Detection 2015*.

BERC Mobile-iris Database - A base de dados BERC é formada por imagens obtidas com infravermelho próximo e possuem resolução de 1280×960 pixels. As imagens foram obtidas por um dispositivo móvel na posição vertical, em sequências compostas de 90 imagens (Kim et al., 2016). A fim de simular situações em que o usuário move o dispositivo para ajustar o foco, as sequências de imagens foram tomadas movendo o telefone móvel em 3 distâncias: 40 para 15 cm, 15 para 25 cm e 25 para 15 cm. As melhores imagens de cada sequência foram selecionadas, totalizando 500 imagens da região periocular de 100 indivíduos. Exemplos de imagens que compõem essa base podem ser visualizados na Figura 4.5.

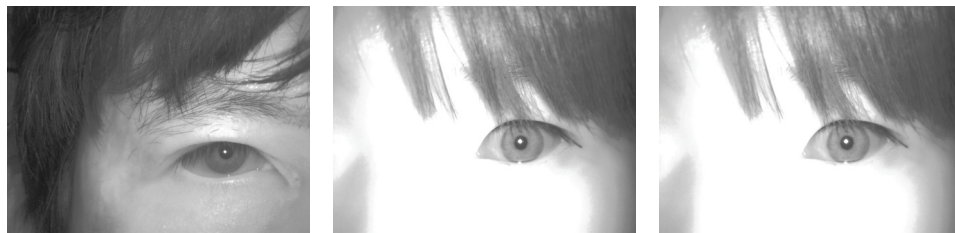


Figura 4.5: Exemplos de imagens encontradas na base de dados BERC.

CASIA-Iris V2 - Essa base de dados é composta por 2400 imagens da região periocular e foi destinada para estudos de características da íris em imagens capturadas em infravermelho próximo. Todas as imagens da base possuem resolução de 640×480 pixels e foram obtidas por meio de dois sensores: OKI IRISPASS-h e CASIA-IrisCamV2. Exemplos de imagens que compõem essa base podem ser visualizados na Figura 4.6.

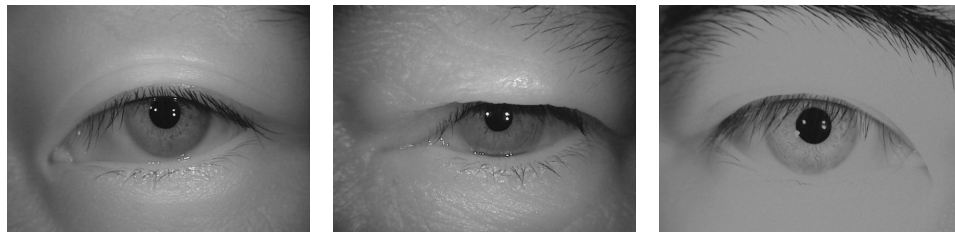


Figura 4.6: Exemplos de imagens encontradas na base de dados *CASIA-IrisV2*

4.1.2 Bases de Imagens VIS

Diversos sistemas biométricos envolvendo íris já foram desenvolvidos e avaliados com imagens obtidas em ambientes controlados de comprimento de onda NIR, em que resultados promissores foram alcançados (Proença e Alexandre, 2011). Por esse motivo, começaram a ser realizadas pesquisas na área de biometria com imagens da íris obtidas em ambientes não controlados e com comprimento de onda visível (VIS) (Proença e Alexandre, 2005; Proença et al., 2009). Atualmente, diversos trabalhos têm sido desenvolvidos em reconhecimento biométrico de íris e região periocular usando imagens obtidas de dispositivos móveis, obtidas em um ambiente não controlado e por diferentes tipos de sensores (De Marsico et al., 2015; Raja et al., 2015; Rattani et al., 2016; Bezerra et al., 2018).

Tabela 4.2: Principais informações das bases de imagens VIS

Base de Dados	Ano	Resumo	Ambiente Controlado	Nº de Imagens
MobBIOfake	2014	Deteção de vivacidade usando imagens de dispositivos móveis	Não	1600
MICHE-I	2015	Reconhecimento de imagens da íris usando imagens de dispositivos móveis	Não	3732

Neste trabalho foram utilizadas duas bases de imagens VIS, sendo elas: MobBIOfake (Sequeira et al., 2014b) e Mobile Iris Challenge Evaluation (MICHE-I) (De Marsico et al., 2015). Na sequência, apresentamos uma descrição e exemplos de imagens dessas bases de dados. Além disso, a Tabela 4.2 apresenta, de forma resumida, as principais características das bases de imagens VIS utilizadas nesse trabalho.

MobBIOfake - A base de dados MobBIOfake foi criada com o propósito de se estudar a detecção de vivacidade (*liveness*) em imagens da íris, obtidas de dispositivos móveis em ambientes não controlados (Sequeira et al., 2014b). Essa base de dados é composta de 1600 imagens da região periocular com resolução de 250×200 pixels, obtidas a partir de um conjunto de 800 imagens pertencentes à base de dados MobBIO (Sequeira et al., 2014a).

Para a construção das imagens falsas, as imagens originais foram agrupadas por indivíduo e um pré-processamento foi realizado, a fim de melhorar o contraste. As imagens foram então impressas utilizando uma impressora profissional em um papel de foto de alta qualidade. Tais imagens foram posteriormente recapturadas utilizando o mesmo dispositivo. Finalmente, as imagens foram cortadas e redimensionadas, a fim de unificar as dimensões. Na Figura 4.7 pode-se observar exemplos de imagens falsas e imagens reais contidas na base MobBIOfake.

Mobile Iris Challenge Evaluation (MICHE-I) - A base de dados MICHE-I consiste em 3732 imagens de 92 indivíduos, obtidas por dispositivos móveis em espectro de luz visível.

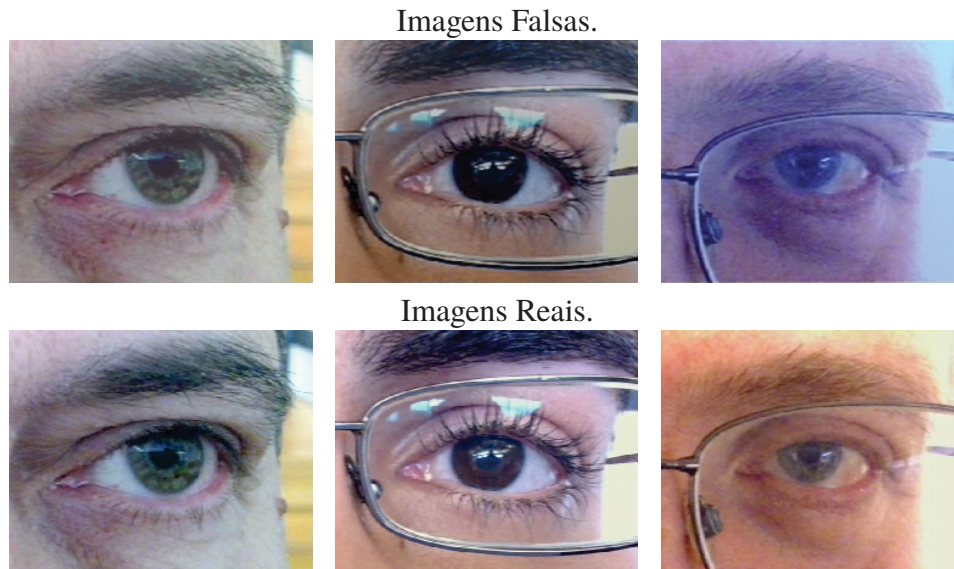


Figura 4.7: Exemplos de imagens encontradas na base de dados *MobBIOfake*.

A fim de simular uma aplicação real, as imagens foram obtidas pelos próprios indivíduos, em ambientes internos e ao ar livre, com e sem óculos.



Figura 4.8: Exemplos de imagens encontradas na base de dados MICHE-I, obtidas com *iPhone 5*.

Para a construção da base de imagens MICHE-I, os dispositivos móveis utilizados e as respectivas resoluções das imagens capturadas foram as seguintes: *iPhone 5* (imagens com resolução de 1536×2048 pixels), *Samsung Galaxy S4* (SGS4) (imagens com resolução de 2322×4128 pixels) e *Samsung Galaxy Tablet II* (SGSII) (imagens com resolução de 640×480 pixels). Devido a forma de aquisição das imagens e ao objetivo da criação dessa base de dados, muitos ruídos são encontrados nas imagens, tais como reflexos, falta de foco, imagens borradas devido a movimentos no momento da aquisição, oclusões, variações na iluminação, dentre outros. Exemplos de imagens contidas nessa base de dados, obtidas através do *iPhone5*, *Samsung Galaxy S4* e *Samsung Galaxy Tablet II* podem ser visualizadas, respectivamente, na Figura 4.8, Figura 4.9 e Figura 4.10.

4.2 MEDIDAS DE AVALIAÇÃO

O nível de exatidão alcançado por métodos de detecção de objetos pode ser quantificado por meio de diversas medidas de avaliação. As medidas de avaliação *Precision* e *Recall* são métricas normalmente utilizadas em detecção de objetos (Lin et al., 2014). Essas métricas são



Figura 4.9: Exemplos de imagens encontradas na base de dados MICHE-I, tomadas com *Samsung Galaxy S4*.



Figura 4.10: Exemplos de imagens encontradas na base de dados MICHE-I, obtidas com *Samsung Galaxy Tablet II*.

calculadas por meio da região do *ground truth* e a região detectada, considerando o número de pixels Falsos Positivos (FPs), Falsos Negativos (FNs), Verdadeiros Positivos (VPs) e Verdadeiros Negativos (VNs) (Davis e Goadrich, 2006), onde:

- *VPs* - Referem-se à exemplos corretamente rotulados como positivos;
- *FPs* - São exemplos negativos incorretamente rotulados como positivos;
- *VNs* - Correspondem aos exemplos corretamente rotulados como negativos;
- *FNs* - são os exemplos positivos incorretamente rotulados como negativos.

Ressaltamos que cada ocorrência de FP, FN, VP ou VN corresponde à um pixel da imagem, como ilustrado na Figura 4.11.

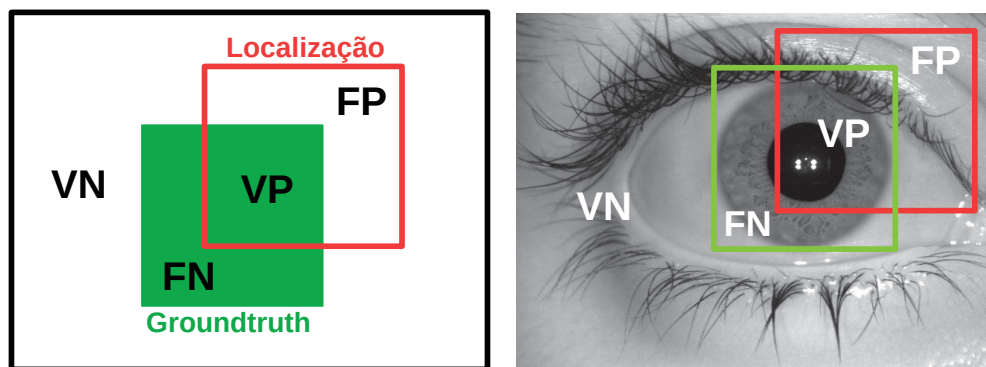


Figura 4.11: Ilustração de pixels Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos, definindo a região de *ground truth* e região detectada

Deste modo, as métricas *Recall* e *Precision* podem ser expressadas, formalmente, da seguinte maneira:

- ***Recall*** - Essa métrica, formalizada na Equação 1, consiste na divisão do número de pixels classificados corretamente como positivos (VP) pela soma entre esse mesmo número de VP's e o número de pixels erroneamente classificados como negativos (FN). Em outras palavras, essa métrica avalia a capacidade de se detectar regiões que fazem parte da região da íris.

$$Recall = \frac{VP}{VP + FN} \quad (1)$$

- ***Precision*** - Na medida de avaliação *precision*, formalizada na Equação 2, quanto maior o número de FPs, menor será o valor resultante de *Precision*. Portanto, a *precision* mede a capacidade que a abordagem possui para não detectar regiões que de fato não devem ser detectadas.

$$Precision = \frac{VP}{VP + FP} \quad (2)$$

Como apontado por (Do et al., 2002), tanto a métrica *Precision* quanto a métrica *Recall*, quando utilizadas isoladamente, não permitem uma avaliação real da qualidade da correspondência entre duas regiões. Em outras palavras, o valor reportado por *Recall* é maximizado em consequência de um baixo número de FNs, que pode ser alcançado por um método de localização pouco rigoroso ao considerar regiões positivas. A baixa rigorosidade do método para determinar regiões positivas provavelmente resultará em um alto número de FPs e, conseqüentemente, em um resultado ruim reportado pela métrica *Precision*. Por outro lado, uma alta precisão pode ser alcançada por meio de um alto limiar de confiança para se considerar regiões positivas, o que poderia gerar, como consequência, um baixo valor de *Recall*.

Em muitos métodos de localização de íris encontrados na literatura, a métrica utilizada para reportar os resultados é a *Accuracy*, formalizada na Equação 3.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (3)$$

Esta métrica é calculada, basicamente, pelo número de acertos (verdadeiros) dividido pelo número total de exemplos (verdadeiros e falsos). Ela é normalmente utilizada em situações com a mesma proporção de exemplos para cada classe. A utilização dessa métrica em problemas com classes desbalanceadas pode causar uma falsa impressão de bom desempenho. Normalmente, em imagens da região periocular, observa-se uma desproporção entre pixels positivos (íris) e negativos (fora da região da íris). Isso ocorre pois a região da íris é, normalmente, significativamente menor do que o restante da imagem, como ilustrado na Figura 4.12.

Uma medida de avaliação frequentemente utilizada para mensurar a qualidade de predições em detecção de objetos consiste na métrica *Intersection over Union* (IoU), também conhecida como Coeficiente de Similaridade de Jaccard. Essa métrica, expressada formalmente na Equação 4, é calculada pela divisão da intersecção entre a área localizada e a área do *ground truth* pela união dessas duas áreas.

$$IoU = \frac{VP}{FP + VP + FN} \quad (4)$$



Figura 4.12: Desbalanceamento de classes observado em imagens da região periocular: Classe positiva (íris) delimitada em verde e classe negativa delimitada em vermelho, ou seja, a região da íris é, geralmente, uma pequena porção da imagem. Imagem obtida da base MICHE-I (De Marsico et al., 2015)

A métrica IoU é interessante porque penaliza tanto localizações superestimadas quanto localizações subestimadas. localizações superestimadas podem incluir uma grande quantidade de informações desnecessárias e também aumentar o tempo de processamento dos estágios subsequentes. Por outro lado, partes significativas do objeto podem ser perdidas em localizações muito rigorosas.

4.3 ABORDAGENS PROPOSTAS

Neste capítulo, as abordagens propostas para o problema de localização de íris são descritas. Na Seção 4.3.1 apresentamos um detector baseado em janela deslizante. Tal abordagem foi implementada com o propósito de criar um *baseline* para detectores de íris. Na Seção 4.3.2, é abordado um detector de íris baseado em aprendizado profundo, fazendo uso de uma versão mais rápida da Darknet YOLO.

4.3.1 Janelas deslizantes com HOG e SVM

Foi proposto um detector baseado em janela deslizante, usando características extraídas por *Histogram of Oriented Gradients* (HOG) e *Support Vector Machine* (SVM) sendo utilizado como classificador, ou seja, uma adaptação do método de detecção de pedestres proposto por Dalal & Triggs (Dalal e Triggs, 2005).

Fazendo uso das imagens de treinamento, é extraída de cada imagem uma janela contendo a região da íris, ou seja, o *ground truth*. Essa janela é utilizada para compor os exemplos de janelas positivas utilizadas para extração de características por meio do HOG. Na Figura 4.13 podemos observar alguns exemplos de janelas positivas.

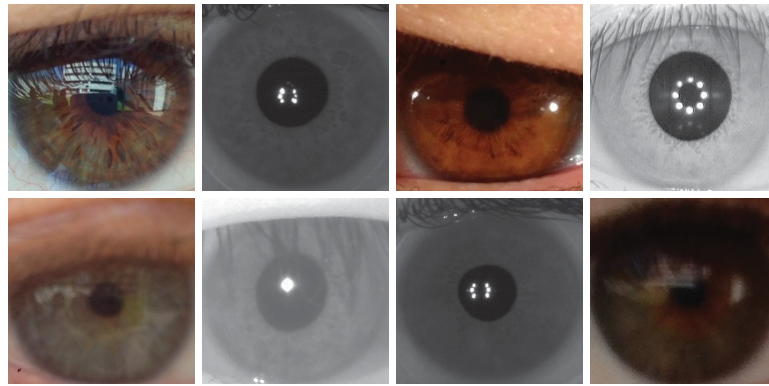


Figura 4.13: Exemplos de amostras positivas utilizadas para extração de características e treinamento do SVM.

Também são extraídas, das imagens de treinamento, janelas contendo regiões que estão totalmente fora na região da íris ou que tem razoável intersecção com a janela que delimita a íris (máximo de 50% da altura ou largura do *ground truth*). Tais regiões são consideradas as janelas negativas, e estão em uma proporção de 20 janelas negativas para cada janela positiva. A Figura 4.14 ilustra exemplos de janelas negativas utilizadas na extração de características.



Figura 4.14: Exemplos de amostras negativas utilizadas para extração de características e treinamento do SVM.

Como parâmetros do HOG, cada janela foi dividida em células de 8×8 pixels. Para cada célula, os gradientes horizontais e verticais em todos os pixels são calculados. Assim, são obtidas as orientações e magnitudes de gradiente. As orientações do gradiente são então quantificadas em nove direções.

A fim de se evitar efeitos de variação de iluminação e contraste, os histogramas de todas as células sobre blocos (2×2 células) foram normalizados. O vetor de características de HOG que descreve cada janela da íris é então construído concatenando os histogramas de células normalizadas para todos os blocos. Finalmente, um vetor de características (2×2 blocos $\times 8$ células $\times 9$ orientações) é obtido para descrever cada janela candidata da íris.

Com as características positivas e negativas extraídas, o classificador SVM pode ser treinado. Para o treinamento no SVM, foi utilizado um *kernel* linear e a constante C é determinada por meio de *grid-search* no conjunto de treinamento.

Para realizar a localização da íris, em cada imagem de teste é aplicada uma abordagem de janela deslizante com diferentes escalas. Adotamos como escala canônica janelas de 150×150 pixels. A partir desta escala, utilizamos 10 escalas inferiores e outras 10 superiores por um fator de 5%. A região da imagem que apresenta maior similaridade com a íris pode ser encontrada por meio da fronteira de decisão gerada pelo SVM. Isso significa que a janela que apresentar maior resposta positiva será considerada como a melhor localização para a íris.

4.3.2 Fast-YOLOv3

Embora tenham sido alcançados excelentes resultados na detecção de objetos usando o YOLOv2, sua arquitetura ainda carece de alguns conceitos essenciais atualmente empregados em diversos detectores estado da arte, tais como blocos residuais, *skip connections* e *upsampling* (Kathuria, 2018). Nesse contexto, Redmon e Farhadi (2018) introduziram a YOLOv3, incluindo os conceitos mencionados acima, bem como um melhor classificador de *backbone*.

Na YOLOv3, uma nova rede, chamada Darknet-53, é usada para extração de características. A Darknet-53 possui 53 camadas convolucionais e é uma abordagem híbrida entre o Darknet-19 e as redes residuais. A função *softmax*, que impõe a suposição de que cada *bounding box* possui exatamente uma classe, foi substituída no YOLOv3 por classificadores independentes, a fim de prever vários rótulos para um objeto. De acordo com Redmon e Farhadi (2018), essa reformulação ajuda ao trabalhar em domínios mais complexos, no qual existem muitos rótulos sobrepostos.

Outra importante característica da YOLOv3 é que são definidas *bounding boxes* em três escalas diferentes, que são dadas precisamente ao reduzir a amostragem das dimensões da imagem de entrada em 32, 16 e 8, respectivamente (Kathuria, 2018). Por exemplo, usando um tamanho de entrada de 416×416 pixels, as *bounding boxes* são definidas nas seguintes escalas: 13×13 , 26×26 e 52×52 pixels. As características são extraídas dessas escalas usando um conceito semelhante ao *Feature Pyramid Networks* (FPNs) (Kathuria, 2018). Observe que as detecções em diferentes camadas ajudam a resolver o problema de detectar objetos pequenos, uma reclamação frequente quanto a YOLO e YOLOv2. A camada 13×13 é responsável pela detecção de objetos grandes, enquanto a camada 26×26 detecta objetos médios e a camada 52×52 detecta objetos menores (Kathuria, 2018).

Como abordagem de aprendizagem profunda aplicada na localização de íris, foi utilizada uma versão mais rápida da YOLOv3, chamada de Fast-YOLOv3 (ou Tiny-YOLOv3). Em comparação com a versão original da YOLO, a Fast-YOLO utiliza um número menor de camadas convolucionais (13 em vez de 24) e menos filtros nessas camadas, como detalhado na Tabela 4.3.

#	Camada	Filtros	Tamanho	Entrada	Saída
0	conv	16	$3 \times 3 \setminus 1$	$416 \times 416 \times 3$	$416 \times 416 \times 16$
1	max		$2 \times 2 \setminus 2$	$416 \times 416 \times 16$	$208 \times 208 \times 16$
2	conv	32	$3 \times 3 \setminus 1$	$208 \times 208 \times 16$	$208 \times 208 \times 32$
3	max		$2 \times 2 \setminus 2$	$208 \times 208 \times 32$	$104 \times 104 \times 32$
4	conv	64	$3 \times 3 \setminus 1$	$104 \times 104 \times 32$	$104 \times 104 \times 64$
5	max		$2 \times 2 \setminus 2$	$104 \times 104 \times 64$	$52 \times 52 \times 64$
6	conv	128	$3 \times 3 \setminus 1$	$52 \times 52 \times 64$	$52 \times 52 \times 128$
7	max		$2 \times 2 \setminus 2$	$52 \times 52 \times 128$	$26 \times 26 \times 128$
8	conv	256	$3 \times 3 \setminus 1$	$26 \times 26 \times 128$	$26 \times 26 \times 256$
9	max		$2 \times 2 \setminus 2$	$26 \times 26 \times 256$	$13 \times 13 \times 256$
10	conv	512	$3 \times 3 \setminus 1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$
11	max		$2 \times 2 \setminus 1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$

#	Camada	Filtros	Tamanho	Entrada	Saída
12	conv	1024	$3 \times 3 \setminus 1$	$13 \times 13 \times 512$	$13 \times 13 \times 1024$
13	conv	256	$1 \times 1 \setminus 1$	$13 \times 13 \times 1024$	$13 \times 13 \times 256$
14	conv	512	$3 \times 3 \setminus 1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$
15	conv	255	$1 \times 1 \setminus 1$	$13 \times 13 \times 512$	$13 \times 13 \times 255$
16	yolo				
17	route				
18	conv	128	$1 \times 1 \setminus 1$	$13 \times 13 \times 256$	$13 \times 13 \times 128$
19	upsample		$2 \times$	$13 \times 13 \times 128$	$26 \times 26 \times 128$
20	route				
21	conv	256	$3 \times 3 \setminus 1$	$26 \times 26 \times 384$	$26 \times 26 \times 256$
22	conv	255	$1 \times 1 \setminus 1$	$26 \times 26 \times 256$	$26 \times 26 \times 255$
23	yolo				

Tabela 4.3: Arquitetura da rede Fast-YOLOv3

5 EXPERIMENTOS

Neste capítulo, apresentamos uma descrição dos experimentos realizados no trabalho, bem como uma discussão dos resultados obtidos. Na Seção 5.1, é descrita a configuração experimental em que a avaliação foi conduzida. Os resultados dos experimentos são apresentados e discutidos na Seção 5.2.

5.1 CONFIGURAÇÃO EXPERIMENTAL

Os experimentos foram feitos em oito bases de imagens (descritas no Capítulo 4) e os resultados foram quantificados por meio das métricas *Recall*, *Precision*, Acurácia e IoU.

A avaliação foi realizada por meio de *Cross-validation* (Validação Cruzada), em que foi empregado o método *K-fold*. O método consiste em dividir a base de imagens em k partes, usando $k - 1$ partes para treino e a parte remanescente para teste, fazendo isso k vezes. Em cada uma das k vezes, testa-se o modelo com um *fold* diferente, calculando a métrica escolhida para avaliação do modelo.

Todos os experimentos foram realizados utilizando uma NVIDIA Titan XP GPU (3840 CUDA cores e 12 GB de RAM) e também fazendo uso de um Intel (R) Core i7-5820K CPU @ 3.30GHz 12 core, 64GB de DDR4 RAM.

5.2 RESULTADOS E DISCUSSÃO

Utilizando a configuração experimental descrita anteriormente, os experimentos são descritos em quatro diferentes cenários: *intra-sensor*, *inter-sensor*, *Combinação de sensores e análise de robustez* da YOLO (experimentos em que a abordagem é testada em uma base não utilizada no conjunto de treinamento).

Tabela 5.1: Resultados *Intra-sensor* - Bases de dados NIR (%)

Database	Recall		Precision		Accuracy		IoU	
	HOG SVM	YOLO	HOG SVM	YOLO	HOG SVM	YOLO	HOG SVM	YOLO
NDCCL								
AD100	93,35	97,98	94,99	95,56	97,27	98,46	88,63	93,60
LG4000	87,96	98,41	94,81	96,07	96,75	98,81	87,74	94,54
IIIT-D CLI								
Vista	90,75	96,35	95,25	96,36	96,73	98,48	86,63	93,36
Cogent	87,99	96,77	96,31	95,98	96,46	98,46	84,52	92,92
BERC	87,82	97,80	99,07	93,63	99,58	99,74	87,04	91,59
CASIA IrisV3								
Interval	88,37	97,65	96,51	95,05	92,82	96,30	86,75	92,82
NDCLD15	90,70	95,98	96,93	97,27	97,36	98,66	88,13	93,59
CASIAV2	76,18	97,82	99,82	95,68	96,42	99,23	76,03	93,59

Tabela 5.2: Resultados *Intra-sensor* - Bases de dados VIS (%)

Database	Recall		Precision		Accuracy		IoU	
	HOG SVM	YOLO	HOG SVM	YOLO	HOG SVM	YOLO	HOG SVM	YOLO
MICHEI								
iPhone 5	82,17	93,87	67,49	94,91	98,28	99,65	60,73	89,06
Samsung Galaxy S4	83,47	95,81	79,26	94,35	98,78	99,72	69,27	90,40
Samsung Galaxy Tab2	80,22	94,65	60,70	91,08	98,05	99,59	54,34	86,30
MobBIO								
Real	90,41	95,55	79,02	95,17	96,56	99,07	71,92	90,99
Fake	76,57	94,92	97,68	94,53	96,16	98,72	74,58	90,91

Resultados *Intra-sensor*: A Tabela 5.1 e Tabela 5.2 apresentam, respectivamente, os resultados obtidos por meio dos experimentos *intra-sensor* em bases NIR e VIS, ou seja, experimentos em que os modelos foram treinados e testados com imagens de um mesmo sensor. A YOLO alcançou as melhores médias em quase todas as métricas analisadas e requer menor tempo de processamento por imagem para a localização da íris. Algumas exceções ocorreram nos experimentos com a base de dados IIIT-D CLI Cogent, CASIAV2 e CASIA-IrisV3 Interval, onde o método baseado em HOG+SVM apresentou melhor resultado na métrica *Precision*. Esse resultado pode ser explicado pelo alto grau de cooperação e controle na aquisição das imagens dessas bases, onde praticamente não há variações de ângulos e todas as imagens foram tomadas de uma distância muito pequena e frontais em relação a íris.

Além disso, é importante destacar que a YOLO localiza a íris em tempo real (entre 0,01 e 0,09 segundos por imagem, em média) utilizando a Titan XP GPU, enquanto a abordagem HOG-SVM demanda, em média, entre 3 e 8 segundos para localizar a íris em cada imagem, utilizando uma única CPU.

Tabela 5.3: Resultados *Inter-sensor* (%)

Database	Set		Recall		Precision		Accuracy		IoU	
	Treinamento	Teste	HOG-SVM	YOLO	HOG-SVM	YOLO	HOG-SVM	YOLO	HOG-SVM	YOLO
NDCCL	AD100	LG4000	90,80	82,55	95,57	91,12	97,14	94,59	87,26	75,49
	LG4000	AD100	87,34	97,54	90,75	93,70	96,05	97,88	83,77	91,33
IIIT-D CLI	Vista	Cogent	85,64	93,52	96,29	96,37	95,83	97,66	82,26	90,24
	Cogent	Vista	92,58	96,66	95,72	95,28	97,22	98,17	88,49	92,18

Resultados *Inter-sensor*: Adicionalmente, para as bases de dados construídas com imagens obtidas com mais de um sensor, experimentos *inter-sensor* foram realizados e são apresentados na Tabela 5.3. Em outras palavras, os detectores foram treinados com imagens do sensor x e testados em imagens do sensor y , pertencentes a mesma base de dados. Por meio desses experimentos, observamos que em alguns casos a YOLO não foi a abordagem que alcançou o melhor resultado. Isso ocorre, por exemplo, na base de dados NDCCL, quando o detector foi treinado com imagens pertencentes ao sensor AD100 e o teste foi realizado com imagens obtidas pelo sensor LG4000.

A razão para o resultado ruim pode estar no fato de que o conjunto de imagens para esse sensor específico (AD100) tem apenas 600 imagens, não permitindo assim uma boa generalização da CNN treinada.

Tabela 5.4: Resultados da combinação de sensores (%)

Database	Conjunto		Recall		Precision		Accuracy		IoU	
	Treinamento	Teste	HOG SVM	YOLO	HOG SVM	YOLO	HOG SVM	YOLO	HOG SVM	YOLO
NDCCL	AD100 & LG4000	LG4000	90,94	97,40	96,30	96,68	97,25	98,60	87,78	94,14
	AD100 & LG4000	AD100	93,20	98,17	95,07	96,73	97,24	98,91	88,57	94,03
IIIT-D CLI	Vista & Cogent	Cogent	86,84	97,10	96,04	95,64	96,09	98,46	83,16	92,90
	Vista & Cogent	Vista	91,49	97,02	94,78	96,37	96,82	98,51	86,93	93,55

Resultados da combinação de sensores: A fim de melhor analisar e entender os resultados dos experimentos *inter-sensor* e confirmar a hipótese de que a performance fraca da YOLO em alguns casos ocorreu devido a poucos exemplos no conjunto de treinamento, experimentos foram realizados combinando os conjuntos de treinamento de diferentes sensores dentro da mesma base de dados.

Os resultados obtidos nesses experimentos podem ser observados na Tabela 5.4. Esses resultados ilustram a importância da diversidade da coleção de imagens utilizadas no treinamento de uma CNN. Por meio de um conjunto de treinamento com maior número de imagens obtidas de diferentes sensores, a YOLO foi capaz de obter maior poder de generalização, melhorando a localização da íris em muitos casos.

Tabela 5.5: Avaliação de robustez da YOLO - Teste em bases desconhecidas (%)

Bases		Recall	Precision	Accuracy	IoU	Tempo
Treinamento	Teste					
{VIS + NIR} - {Teste}	MICHE iPhone 5	94,13 ± 4,99	95,92 ± 3,82	99,70 ± 0,17	90,33 ± 4,41	0,05 s
	IIIT-D Vista	95,54 ± 3,26	96,39 ± 2,73	98,19 ± 1,39	92,18 ± 3,33	0,02 s
	IIIT-D Cogent	95,96 ± 4,81	94,50 ± 3,95	97,93 ± 1,64	90,71 ± 4,50	0,02 s
	MICHE SGS4	94,42 ± 4,95	94,85 ± 4,99	99,70 ± 0,29	89,60 ± 4,97	0,09 s
	BERC	96,99 ± 2,99	93,58 ± 5,23	99,61 ± 0,57	91,04 ± 4,73	0,02 s
	MICHE SGTII	97,47 ± 3,48	90,31 ± 5,43	99,62 ± 1,12	88,10 ± 4,86	0,02 s
	NDCCL AD100	95,51 ± 2,89	96,60 ± 4,62	98,23 ± 1,09	92,30 ± 4,36	0,02 s
	NDCCL LG4000	93,64 ± 4,77	97,35 ± 3,92	98,13 ± 1,27	91,22 ± 4,88	0,02 s
	MobBio Real	95,39 ± 3,92	95,18 ± 3,87	98,85 ± 1,48	90,40 ± 3,92	0,01 s
	MobBio Fake	94,63 ± 3,25	94,18 ± 3,41	98,64 ± 1,55	90,27 ± 3,31	0,01 s
	Casia IrisV2	95,47 ± 2,73	94,02 ± 3,36	97,75 ± 1,65	92,66 ± 2,94	0,01 s

Análise de robustez da YOLO: A Tabela 5.5 contém os resultados obtidos por meio de experimentos em que a YOLO foi treinada com imagens de todas as bases, exceto as imagens da base em que seria testada, ou seja, seria testada em bases de imagens desconhecidas. Nesses experimentos, observamos que as métricas reportaram um comportamento seguro da YOLO (IoU de pelo menos 90% na maioria das bases), mesmo sendo testada em bases não vistas no treinamento.

Adicionalmente, em alguns casos (quando testada nas bases MICHE Iphone 5, por exemplo) os resultados foram até superiores aos obtidos em experimentos anteriores (experimentos em que haviam imagens da mesma base tanto no treinamento quanto no teste). Isso provavelmente ocorreu pois, como o conjunto de treinamento é formado com imagens de todas as bases (exceto as imagens da base em que será realizado o teste), tem-se uma maior variedade e volume de imagens para o treino. Com isso, a YOLO melhora sua capacidade de generalização, podendo apresentar resultados até melhores que os reportados nos experimentos anteriores.

Tais experimentos evidenciam a confiabilidade da utilização da YOLO na detecção de íris, mesmo em bases de imagens não utilizadas no treinamento. A Figura 5.1 apresenta exemplos de detecções realizadas nesse cenário (bases desconhecidas), em imagens com significativa presença de ruídos, tais como óculos, sobrancelha, variação de iluminação, oclusões, dentre outros.

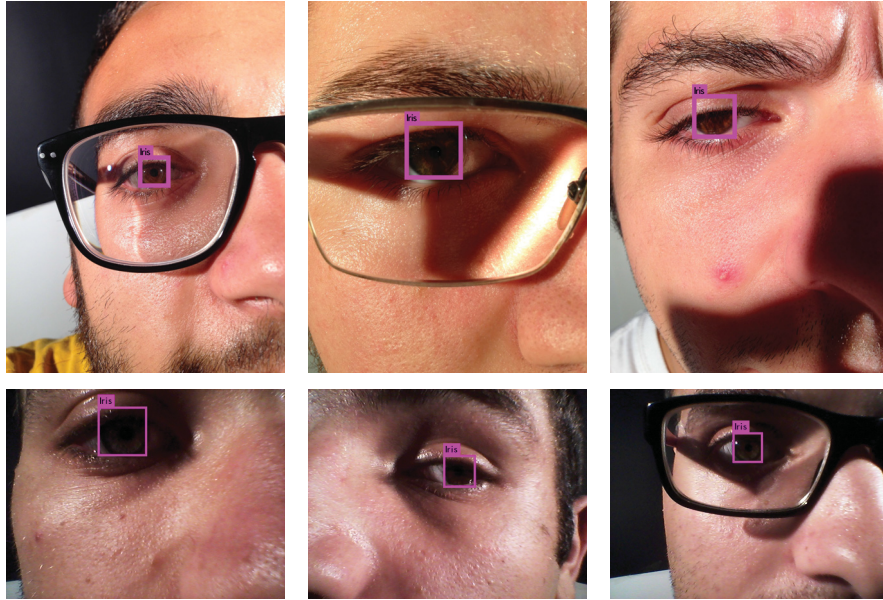


Figura 5.1: Exemplos de detecções (em bases desconhecidas) em imagens com significativa presença de ruídos

As Figuras 5.2 e 5.3 ilustram o comportamento da curvas das métricas *recall*, *precision* e *IoU* para os experimentos reportados na Tabela 5.5. Os gráficos mostram como a porcentagem de imagens (quantidade de imagens) varia de acordo com os níveis de cada métrica. Por meio dessa análise, observa-se que a YOLO é uma alternativa robusta e promissora para a localização de íris, visto que, mesmo com a abordagem sendo testada em bases "desconhecidas", em mais de 90% das imagens foi reportado um valor de IoU superior à 80%. Tal observação também é válida para as métricas *recall* e *precision*. Na Figura 5.4(a), pode-se observar um exemplo de detecção em que foi reportado IoU de aproximadamente 80%. Outros exemplos de detecções com seus respectivos valores de IoU podem ser visualizados na Figuras 5.4(b) e 5.4(c).

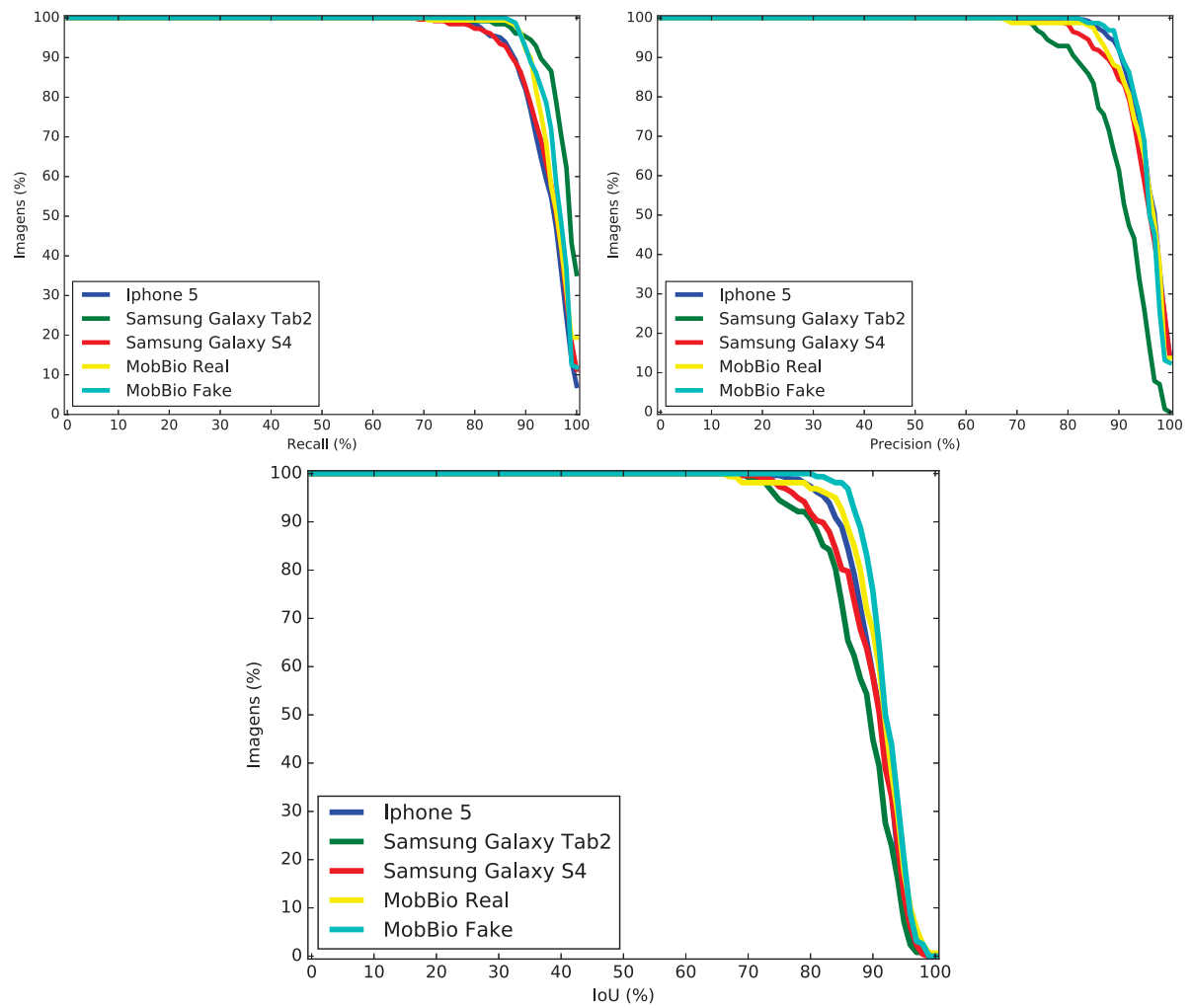


Figura 5.2: Curvas das métricas Recall, Precision e IoU para os resultados obtidos pela YOLO em experimentos com bases desconhecidas VIS (Tabela 5.5).

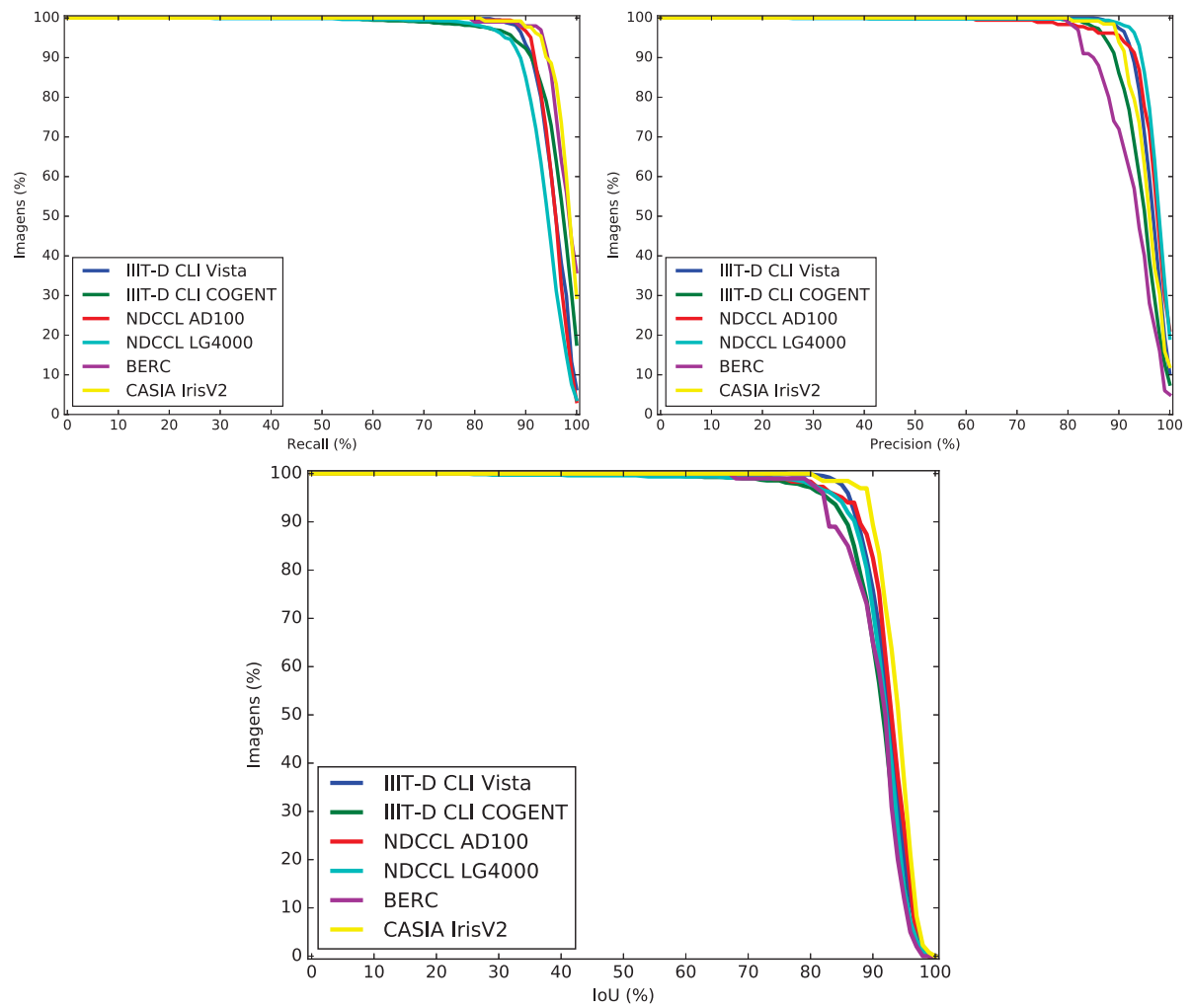


Figura 5.3: Curvas das métricas Recall, Precision e IoU para os resultados obtidos pela YOLO em experimentos com bases desconhecidas NIR (Tabela 5.5).

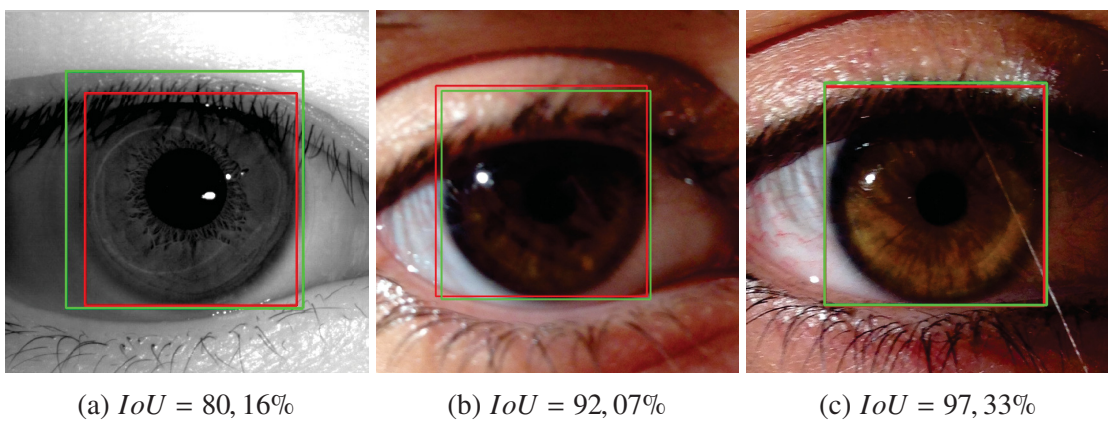


Figura 5.4: Exemplos de detecções obtidas pela YOLO e seus respectivos valores de IoU. A *bounding box* verde refere-se ao *ground truth* e a *bounding box* vermelha refere-se à região detectada.

6 CONCLUSÕES

A íris aparece como uma das mais representativas e seguras modalidades biométricas, pois consiste em uma região do olho humano que permite distinguir indivíduos com alto grau de unicidade (Zhu et al., 2000). Além disso, o processo de identificação não é invasivo, ou seja, não há a necessidade de contato físico com a íris para que a imagem seja obtida e analisada (Jain et al., 2006).

Em sistemas de reconhecimento, autenticação e identificação de pessoas por meio da íris, a localização da região da íris geralmente é um dos passos iniciais e pode, conseqüentemente, influenciar diretamente o desempenho de tais sistemas. Neste trabalho, duas abordagens de detecção de objetos são avaliadas na localização de íris, sendo elas: 1) Um detector baseado em janela deslizante, usando características extraídas por HOG e fazendo uso do SVM como classificador; 2) Um detector baseado em *deep learning*, utilizando a CNN Darknet YOLO.

As abordagens foram avaliadas em oito bases de imagens da região periocular. Foram manualmente anotadas seis bases de dados das oito utilizadas. Tais anotações estão disponíveis publicamente para serem utilizadas em fins científicos. Além disso, a avaliação ocorreu por meio de quatro diferentes cenários: *intra-sensor*, *inter-sensor*, *combinação de sensores* e *análise de robustez da YOLO*. Os resultados foram reportados pelas métricas *recall*, *precision*, *accuracy* e *IoU*.

Os experimentos mostram que o uso do detector de objetos YOLO, baseado em *Deep Learning*, aplicado para a localização de íris apresenta resultados promissores em todas as bases de dados utilizadas. Mesmo no cenário de experimentos em que a abordagem foi testada em bases "desconhecidas" (bases não incluídas no conjunto de treinamento), em mais de 90% das imagens testadas foi reportado um valor de *IoU* superior à 80%. Tal observação também é válida para as métricas *recall* e *precision*.

Além disso, a localização da íris por meio dessa abordagem ocorre em tempo real (entre 0,02 e 0,09 segundos - dependendo da base de dados - por imagem, em média), utilizando uma GPU (NVIDIA GeForce Titan XP Pascal).

Outra importante conclusão a ser mencionada até agora é que, igualmente a outras abordagens de *Deep Learning*, um número de imagens suficientemente grande é necessário para o treinamento. O número e variedade de imagens no conjunto de treinamento afeta diretamente a capacidade de generalização do modelo treinado.

Para a continuidade deste trabalho, pretende-se avaliar o impacto causado pela detecção da íris em sistemas de reconhecimento. Também pretende-se investigar a otimização da arquitetura da rede de convolução utilizada para detecção da íris, de modo a obter um detector de menor custo computacional e que alcance maior exatidão na localização da íris. Adicionalmente, pretende-se analisar a utilização de outras abordagens de redes neurais, tais como *Single Shot Detector* (SSD) e *Faster R-CNN*.

6.1 PUBLICAÇÕES

Os trabalhos publicados durante o mestrado estão listados abaixo. As publicações relacionadas à dissertação estão marcadas com asterisco [*]. Embora alguns artigos não estejam diretamente relacionados à detecção de íris, abordagens de aprendizagem profunda (incluindo modelos da YOLO) foram empregadas em todos eles, contribuindo para os resultados obtidos neste trabalho.

- [*] E. Severo, R. Laroca, C. S. Bezerra, L. A. Zanlorensi, D. Weingaertner, G. Moreira, D. Menotti. (2018). A Benchmark for Iris Location and a Deep Learning Detector Evaluation. *International Joint Conference on Neural Networks (IJCNN)*.
- [*] E. Severo. e D. Menotti. (2017). A comparison of Texture and Shape Descriptors for Iris Location Based on a Sliding Window Approach. *XIII Workshop de Visão Computacional (WVC)*.
- R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, D. Menotti. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. *International Joint Conference on Neural Networks (IJCNN)*.
- D. R. Lucio, R. Laroca, E. Severo, A. S. Britto Jr., D. Menotti. (2018). Fully Convolutional Networks and Generative Adversarial Networks Applied to Sclera Segmentation. *IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*.
- C. S. Bezerra, R. Laroca, D. R. Lucio, E. Severo, L. F. Oliveira, A. S. Britto Jr., D. Menotti. (2018). Robust Iris Segmentation Based on Fully Convolutional Networks and Generative Adversarial Networks. *Conference on Graphics, Patterns and Images (SIBGRAPI)*.

REFERÊNCIAS

- Ahuja, K., Islam, R., Barbhuiya, F. A. e Dey, K. (2017). Convolutional neural networks for ocular smartphone-based biometrics. *Pattern Recognition Letters*, 91:17–26.
- Alvarez, Y. e Garcia, M. (2010). A fast iris location based on aggregating gradient approximation using qma-owa operator. Em *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, páginas 1–8. IEEE.
- Arora, S. S., Vatsa, M., Singh, R. e Jain, A. (2012). Iris recognition under alcohol influence: A preliminary study. Em *Biometrics (ICB), 2012 5th IAPR International Conference on*, páginas 336–341. IEEE.
- Baker, S. E., Bowyer, K. W., Flynn, P. J. e Phillips, P. J. (2013). Template aging in iris biometrics. Em *Handbook of Iris Recognition*, páginas 205–218. Springer.
- Baker, S. E., Hentz, A., Bowyer, K. W. e Flynn, P. J. (2010). Degradation of iris recognition performance due to non-cosmetic prescription contact lenses. *Computer vision and image understanding*, 114(9):1030–1044.
- Bengio, Y., Courville, A. e Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.
- Bezerra, C. S., Laroca, R., Lucio, D. R., Severo, E., Oliveira, L. F., Britto Jr., A. S. e Menotti, D. (2018). Robust iris segmentation based on fully convolutional networks and generative adversarial networks. Em *Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 281–288.
- Bhilaré, S., Kanhangad, V. e Chaudhari, N. (2017). Histogram of oriented gradients based presentation attack detection in dorsal hand-vein biometric system. Em *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, páginas 39–42. IEEE.
- Bowyer, K. W., Hollingsworth, K. e Flynn, P. J. (2008). Image understanding for iris biometrics: A survey. *Computer vision and image understanding*, 110(2):281–307.
- CASIA (2010). Casia-irisv3 image database center for biometrics and security research (cbsr). <http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp>. Acessado em 20/11/2017.
- CASIA-IrisV2 (2004). CASIA Iris Image Database Version 2.0 (CASIA-Iris V2). <http://http://biometrics.idealtest.org/dbDetailForUser.do?id=2>.
- Chauhan, S., Arora, A. e Kaul, A. (2010). A survey of emerging biometric modalities. *Procedia Computer Science*, 2:213–218.
- Crihalmeanu, S., Ross, A., Schuckers, S. e Hornak, L. (2007). A protocol for multibiometric data acquisition, storage and dissemination. *Technical Report, WVU, Lane Department of Computer Science and Electrical Engineering*.

- Cui, W. et al. (2012). A rapid iris location algorithm based on embedded. Em *Computer Science and Information Processing (CSIP), 2012 International Conference on*, páginas 233–236. IEEE.
- Czajka, A. (2013). Database of iris printouts and its application: Development of liveness detection method for iris recognition. Em *Methods and Models in Automation and Robotics (MMAR), 2013 18th International Conference on*, páginas 28–33. IEEE.
- Dalal, N. e Triggs, B. (2005). Histograms of oriented gradients for human detection. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 886–893. IEEE.
- Daugman, J. (2004). How iris recognition works. *IEEE Transactions on circuits and systems for video technology*, 14(1):21–30.
- Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1148–1161.
- Davis, J. e Goadrich, M. (2006). The relationship between precision-recall and roc curves. Em *Proceedings of the 23rd international conference on Machine learning*, páginas 233–240. ACM.
- De Marsico, M., Nappi, M., Riccio, D. e Wechsler, H. (2015). Mobile iris challenge evaluation (miche)-i, biometric iris dataset and protocols. *Pattern Recognition Letters*, 57:17–23.
- Do, H.-H., Melnik, S. e Rahm, E. (2002). Comparison of schema matching evaluations. Em *Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World*, páginas 221–237. Springer.
- Doyle, J. e Bowyer, K. (2014). Notre dame image database for contact lens detection in iris recognition—2013.
- Doyle, J. S. e Bowyer, K. W. (2015). Robust detection of textured contact lenses in iris recognition using bsif. *IEEE Access*, 3:1672–1683.
- Doyle, J. S., Bowyer, K. W. e Flynn, P. J. (2013). Variation in accuracy of textured contact lens detection based on sensor and lens pattern. Em *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, páginas 1–7. IEEE.
- Duda, R. O., Hart, P. E. e Stork, D. G. (2003). *Pattern classification*.
- Dumoulin, V. e Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.
- Fan, T.-Y., Mu, Z.-C. e Yang, R.-Y. (2017). Multi-modality recognition of human face and ear based on deep learning. Em *Wavelet Analysis and Pattern Recognition (ICWAPR), 2017 International Conference on*, páginas 38–42. IEEE.
- Fenker, S. P. e Bowyer, K. W. (2012). Analysis of template aging in iris biometrics. Em *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, páginas 45–51. IEEE.

- Franchi, G., Angulo, J. e Sejdinović, D. (2016). Hyperspectral image classification with support vector machines on kernel distribution embeddings. Em *Image Processing (ICIP), 2016 IEEE International Conference on*, páginas 1898–1902. IEEE.
- Fukushima, K. e Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. Em *Competition and cooperation in neural nets*, páginas 267–285. Springer.
- Girshick, R. (2015). Fast r-cnn. *arXiv preprint arXiv:1504.08083*.
- Goodfellow, I., Bengio, Y. e Courville, A. (2016). *Deep learning*. MIT press.
- Gupta, P., Behera, S., Vatsa, M. e Singh, R. (2014). On iris spoofing using print attack. Em *Pattern Recognition (ICPR), 2014 22nd International Conference on*, páginas 1681–1686. IEEE.
- Hinton, G. E. e Salakhutdinov, R. R. (2016). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.
- Hui, J. (2018). Real-time object detection with YOLO, YOLOv2 and now YOLOv3. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088.
- Jain, A., Bolle, R. e Pankanti, S. (2006). *Biometrics: personal identification in networked society*, volume 479. Springer Science & Business Media.
- Jain, A. K., Duin, R. P. W. e Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37.
- Jain, A. K., Ross, A. e Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1):4–20.
- Jillela, R., Ross, A. A., Boddeti, V. N., Kumar, B. V., Hu, X., Plemmons, R. e Pauca, P. (2013). Iris segmentation for challenging periocular images. Em *Handbook of Iris Recognition*, páginas 281–308. Springer.
- Kathuria, A. (2018). What's new in yolo v3? <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>.
- Kim, D., Jung, Y., Toh, K.-A., Son, B. e Kim, J. (2016). An empirical study on iris recognition in a mobile phone. *Expert Systems with Applications*, 54:328–339.
- Kohli, N., Yadav, D., Vatsa, M. e Singh, R. (2013). Revisiting iris recognition with color cosmetic contact lenses. Em *Biometrics (ICB), 2013 International Conference on*, páginas 1–7. IEEE.
- Kohli, N., Yadav, D., Vatsa, M., Singh, R. e Noore, A. (2016). Detecting medley of iris spoofing attacks using desist. Em *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, páginas 1–6. IEEE.
- Krizhevsky, A., Sutskever, I. e Hinton, G. E. (2012a). ImageNet Classification with Deep Convolutional Neural Networks. Em *Advances in Neural Information Processing Systems*.

- Krizhevsky, A., Sutskever, I. e Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. Em *Advances in neural information processing systems*, páginas 1097–1105.
- Laroca, R., Barroso, V., Diniz, M. A., Gonçalves, G. R., Schwartz, W. R. e Menotti, D. (2019a). Convolutional neural networks for automatic meter reading. *Journal of Electronic Imaging*, 28:1–14.
- Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R. e Menotti, D. (2018). A robust real-time automatic license plate recognition based on the YOLO detector. Em *International Joint Conference on Neural Networks (IJCNN)*, páginas 1–10.
- Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R. e Menotti, D. (2019b). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *arXiv preprint*, arXiv:1909.01754:1–14.
- Le, C. e Jain, R. (2009). A survey of biometrics security systems. *EEUU. Washington University in St. Louis*.
- LeCun, Y., Bengio, Y. e Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.
- LeCun, Y., Bottou, L., Orr, G. B. e Müller, K.-R. (1998). Efficient backprop. Em *Neural networks: Tricks of the trade*, páginas 9–50. Springer.
- Li, Y., Li, W. e Ma, Y. (2012). Accurate iris location based on region of interest. Em *Biomedical Engineering and Biotechnology (iCBEB), 2012 International Conference on*, páginas 704–707. IEEE.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. e Zitnick, C. L. (2014). Microsoft coco: Common objects in context. Em *European conference on computer vision*, páginas 740–755. Springer.
- Lucio, D. R., Laroca, R., Severo, E., Britto Jr., A. S. e Menotti, D. (2018). Fully convolutional networks and generative adversarial networks applied to sclera segmentation. Em *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, páginas 1–7.
- Mao, Z., Yao, W. X. e Huang, Y. (2017). Eeg-based biometric identification with deep learning. Em *Neural Engineering (NER), 2017 8th International IEEE/EMBS Conference on*, páginas 609–612. IEEE.
- Menotti, D., Chiachia, G., Pinto, A., Schwartz, W. R., Pedrini, H., Falcao, A. X. e Rocha, A. (2015). Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Transactions on Information Forensics and Security*, 10(4):864–879.
- Oad, P. e Ahmad, W. (2012). Iris localization using daugman’s algorithm.
- Patil, P. S. (2014). Research on iris region localization algorithms. *Int. Journal of Engineering Research and Applications www.ijera.com ISSN*, páginas 2248–9622.
- Peláez, J. I. e Doña, J. M. (2006). A majority model in group decision making using qma–owa operators. *International Journal of Intelligent Systems*, 21(2):193–208.

- Phillips, P. J., Bowyer, K. W., Flynn, P. J., Liu, X. e Scruggs, W. T. (2008). The iris challenge evaluation 2005. Em *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, páginas 1–8. IEEE.
- Phillips, P. J., Scruggs, W. T., O’Toole, A. J., Flynn, P. J., Bowyer, K. W., Schott, C. L. e Sharpe, M. (2007). Frvt 2006 and ice 2006 large-scale results. *National Institute of Standards and Technology, NISTIR*, 7408(1).
- Pinto, N., Stone, Z., Zickler, T. e Cox, D. D. (2011). Scaling-up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. Em *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, páginas 35–42.
- Proença, H. e Alexandre, L. A. (2005). Ubiris: A noisy iris image database. Em *International Conference on Image Analysis and Processing*, páginas 970–977. Springer.
- Proença, H. e Alexandre, L. A. (2006). Iris segmentation methodology for non-cooperative recognition. *IEE Proceedings-Vision, Image and Signal Processing*, 153(2):199–205.
- Proenca, H., Filipe, S., Santos, R., Oliveira, J. e Alexandre, L. A. (2009). The ubiris. v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1529–1535.
- Proença, H. e Alexandre, L. A. (2011). Toward covert iris biometric recognition: Experimental results from the nice contests. *IEEE Transactions on Information Forensics and Security*, 7(2):798–808.
- Raghavendra, C., Kumaravel, A. e Sivasubramanian, S. (2017). Iris technology: A review on iris based biometric systems for unique human identification. Em *Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), 2017 International Conference on*, páginas 1–6. IEEE.
- Raja, K. B., Raghavendra, R., Vemuri, V. K. e Busch, C. (2015). Smartphone based visible iris recognition using deep sparse filtering. *Pattern Recognition Letters*, 57:33–42.
- Rattani, A., Derakhshani, R., Saripalle, S. K. e Gottemukkula, V. (2016). Icip 2016 competition on mobile ocular biometric recognition. Em *2016 IEEE International Conference on Image Processing (ICIP)*, páginas 320–324. IEEE.
- Redmon, J. (2013–2016a). Darknet: Open source neural networks in C. <http://pjreddie.com/darknet/>.
- Redmon, J. (2016b). Yolo: Real-time object detection. <https://pjreddie.com/darknet/>. Acessado em 02/05/2018.
- Redmon, J., Divvala, S., Girshick, R. e Farhadi, A. (2016a). You only look once: Unified, real-time object detection. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 779–788.
- Redmon, J., Divvala, S., Girshick, R. e Farhadi, A. (2016b). You only look once: Unified, real-time object detection. Em *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 779–788.

- Redmon, J. e Farhadi, A. (2017). YOLO9000: better, faster, stronger. Em *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Redmon, J. e Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R. e Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Em *Advances in neural information processing systems*, páginas 91–99.
- Rodríguez, J. e Rubio, Y. (2005). A new method for iris pupil contour delimitation and its application in iris texture parameter estimation. *Progress in Pattern Recognition, Image Analysis and Applications*, páginas 631–641.
- Ruiz, P., Mateos, J., Camps-Valls, G., Molina, R. e Katsaggelos, A. K. (2014). Bayesian active remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):2186–2196.
- Ruiz-Albacete, V., Tome-Gonzalez, P., Alonso-Fernandez, F., Galbally, J., Fierrez, J. e Ortega-Garcia, J. (2008). Direct attacks using fake images in iris verification. Em *European Workshop on Biometrics and Identity Management*, páginas 181–190. Springer.
- Sequeira, A. F., Monteiro, J. C., Rebelo, A. e Oliveira, H. P. (2014a). Mobbio: a multimodal database captured with a portable handheld device. Em *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, páginas 133–139. IEEE.
- Sequeira, A. F., Murari, J. e Cardoso, J. S. (2014b). Iris liveness detection methods in mobile applications. Em *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 3, páginas 22–33. IEEE.
- Severo, E., Laroca, R., Bezerra, C. S., Zanolensi, L. A., Weingaertner, D., Moreira, G. e Menotti, D. (2018). A benchmark for iris location and a deep learning detector evaluation. páginas 1–7.
- Severo, E. e Menotti, D. (2017). A comparison of texture and shape descriptors for iris location based on a sliding window approach. Em *XIII Workshop de Visão Computacional (WVC 2017)*.
- Shah, S. e Ross, A. (2006). Generating synthetic irises by feature agglomeration. Em *Image Processing, 2006 IEEE International Conference on*, páginas 317–320. IEEE.
- Simonyan, K. e Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*.
- Su, L., Wu, J., Li, Q. e Liu, Z. (2017). Iris location based on regional property and iterative searching. Em *Mechatronics and Automation (ICMA), 2017 IEEE International Conference on*, páginas 1064–1068. IEEE.
- Sundaram, R. M., Dhara, B. C. e Chanda, B. (2011). A fast method for iris localization. Em *Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on*, páginas 89–92. IEEE.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. e Rabinovich, A. (2015). Going deeper with convolutions. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1–9.

- Tapia, J. E., Perez, C. A. e Bowyer, K. W. (2016). Gender classification from the same iris code used for recognition. *IEEE Transactions on Information Forensics and Security*, 11(8):1760–1770.
- Theodoridis, S. e Koutroumbas, K. (2003). *Pattern Recognition*. John Wiley & Sons.
- Turk, M. e Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86.
- Vapnik, V. N. (1995). The nature of statistical learning. *Theory*.
- Vargas, A. C. G., Paes, A. e Vasconcelos, C. N. (2016). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. Em *Proceedings of the XXIX Conference on Graphics, Patterns and Images*, páginas 1–4.
- Wang, X., Han, T. X. e Yan, S. (2009). An hog-lbp human detector with partial occlusion handling. Em *Computer Vision, 2009 IEEE 12th International Conference on*, páginas 32–39. IEEE.
- Wildes, R. P. (1997). Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363.
- Xiaorong, X. e Yongjun, L. (2011). A fast iris location method based on combined coarse and fine treatment. *Journal of Lanzhou University of Technology*.
- Yadav, D., Kohli, N., Doyle, J. S., Singh, R., Vatsa, M. e Bowyer, K. W. (2014). Unraveling the effect of textured contact lenses on iris recognition. *IEEE Transactions on Information Forensics and Security*, 9(5):851–862.
- Zhang, W. e Ma, Y.-D. (2014). A new approach for iris localization based on an improved level set method. Em *Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2014 11th International Computer Conference on*, páginas 309–312. IEEE.
- Zhou, L., Ma, Y., Lian, J. e Wang, Z. (2013). A new effective algorithm for iris location. Em *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, páginas 1790–1795. IEEE.
- Zhu, Y., Tan, T. e Wang, Y. (2000). Biometric personal identification based on iris patterns. Em *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, páginas 801–804. IEEE.
- Zuo, J., Schmid, N. A. e Chen, X. (2007). On generation and analysis of synthetic iris images. *IEEE Transactions on Information Forensics and Security*, 2(1):77–90.